1988

Multivariable Adaptive Control Design Under Internal Model Control Structure.

Ivan Patricio Solar
Louisiana State University and Agricultural & Mechanical College

INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the original text directly from the copy submitted. Thus, some dissertation copies are in typewriter face, while others may be from a computer printer.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyrighted material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is available as one exposure on a standard 35 mm slide or as a 17" × 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. 35 mm slides or 6" × 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Accessing the World's Information since 1938

300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

Order Number 8819983

Multivariable adaptive control design under internal model control structure

Solar, Ivan Patricio, Ph.D.

The Louisiana State University and Agricultural and Mechanical Col., 1988

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __✓__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy __✓__
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages __✓__
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Dissertation contains pages with print at a slant, filmed as received __✓__
16. Other _____

_____

_____

MULTIVARIABLE ADAPTIVE CONTROL DESIGN

UNDER INTERNAL MODEL CONTROL STRUCTURE

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Chemical Engineering

by
Ivan Patricio Solar
B.S. University of Chile 1971
DIC Imperial College 1974
M.S. University of London 1975
May 1988

ACKNOWLEDGMENTS

This work was developed under the guidance of Dr. F. R. Groves Jr. to whom I want to express my gratitude for his helpful suggestions and encouragement.

Thanks are also due to Dr. J. Aravena for his advice and friendship.

My appreciation to the members of the examining committee, Drs. A. B. Corripio, R. Pike, D. Reible and R. G. Hussey.

Dr. Armando Corripio deserves a special acknowledgment for his help with the computer system and the subroutines he has created.

I am also deeply grateful to Dr. R. Perez who taught me about the fast estimation methods.

Thanks are given to LSU for providing the economic support I received during these years. I extend my gratitude to Pontificia Universidad Catolica de Chile.

Finally, I want to express my gratitude and love to my dear wife Isabel and lovely daughter Katia for their sacrifice and love.

Once more, Thanks!

TABLE OF CONTENTS

LIST OF SYMBOLS

A(z):Estimated polynomial matrix of order N_{AM}.Represents the output matrix.

B(z):Estimated polynomial matrix of order N_{BM}.Represents the input matrix.

C :Constant coefficients matrix associated with the output vector

C :Field of complex numbers

d :Deterministic disturbance

$\hat{d}$:Estimated disturbance

D :Diagonal matrix used to construct the best inverse of B

e :Estimation error

E :Expected value operator

F :Filter matrix;diagonal with first order lags in this work

G(s):Process transfer function matrix

Gc :Control transfer function matrix

G_+ :Non invertible factor of G

G_- :Invertible factor of G

G# :Estimated model of the plant

G_K :Adaptive gain schedule matrix (for nonlinear systems)

h :Sampling time period

I :Identity matrix

k :Discrete time

m :Number of inputs and outputs

N :Degree of det(B)

N_{AM} :Model output matrix degree

N_{BM} :Model input matrix degree

ND :Global dead time of the model

N^+ :Degree of the unstable part of det(B)

N^- :Degree of the stable part of det(B)

NPM :Total number of model parameters

N_r :Degree of matrix B(z) inverse

p :Generalized variable for discrete or continuous systems

pij :Minimum delay of numerator of βij

P :Information matrix

qij :Minimum delay of denominator of βij

Q :Square root of matrix P

R :Field of real numbers

R_a :Ring of proper stable scalar transfer functions

R_a^{mxn} :Ring of (mxn) matrices with elements in R

s :Laplace transform variable

sp :Set point vector

t :Time

u :Control vector

x :state vector

y :Output vector

$\hat{y}$:Estimated Output

y_m :Model output

z :Discrete transform variable

Greek Letters

α_i :Filter tuning parameters

β_{ij}:Elements of matrix B inverse

γ :Condition number defined by σ_M / σ_m

δ :Column vector added to θ matrix

Δ :Det(B)

Δ^+ :Unstable factor of det(B)

Δ^- :Stable factor of det(B)

ε :Generalized error for internal model control

ϕ :regressor vector for estimation

λ :Variable forgetting factor

Λ :Unstable region

ρ_i :Eigenvalues

σ :Variance

σ_M :Greatest singular value

σ_m :Minimum singular value

σ_o :Window estimation factor

θ :Matrix of estimated parameters

ψ_i :Elements of Smith form matrix

Acronyms

ACS :Advanced Control System

ARX :Auto Regressive model with external forcing function

BIBO :Bounded Input Bounded Output

CSMP :Continuous Systems Modeling Program

ERLS :Extended Recursive Least Square

IBM :Interntional Business Machines

IMC :Internal Model Control

LINPACK:Linear Package

LQG :Linear Quadratic Gaussian

MIMO :Multiple Input Multiple Output

MIMO :Mean Time Between Failure

PRBS :Pseudo Random Binary Sequence

RLS :Recursive Least Square

SAS :Statistical Analysis System

STC :Self Tuning Controller

STR :Self Tuning Regulator

WNS :White Noise Sequence

LIST OF FIGURES

ABSTRACT

A new adaptive multivariable control scheme has been devised. The method combines the best characteristics of conventional adaptive systems and internal model control (IMC) structure. The control scheme builds by itself the required models and avoids the ambiguities in the definition of performance specifications.

The problem of plant inversion associated with the IMC structure has been solved. The method introduced in this work is based on the properties of the Smith-McMillan form. However, the method does not require the explicit determination of the form. Furthermore, the computation of a stable plant inverse requires only matrix inversion and scalar polynomial factorization. The resulting algorithm is suitable for on-line operation.

The control scheme is implemented through the following stages:

1.- Identification

The parameters of a multivariable ARX model are estimated using a recursive least square algorithm with variable forgetting factor. The input and output orders can be used as additional degrees of freedom. The algorithm developed shows good numerical characteristics with fast convergence even

for a large number of parameters.

2.- Computation of the Manipulated Variables

The model is used to determine a controller following the IMC approach. The resulting equations are solved to compute the required manipulated variables. The algorithm for system inversion allows computations to be executed on-line.

3.- Filtering

The usual filters of the IMC approach are also used in the adaptive scheme. The objective is to reduce the sensitivity of the controller. Only non-adaptive non-interactive filters have been considered. The results with first order low pass filters are satisfactory. The bandwidth of the filter is used as an additional tuning parameter.

The adaptive control strategy has been extensively tested using computer simulation. The tests include extensions to non-linear plants. Comparisons with non-adaptive IMC control show the advantage of the new scheme developed in this work.

CHAPTER I

INTRODUCTION

The design of a control system for a multivariable chemical process is one of the most challenging tasks that a control engineer can confront. In general, the demands made on the controller are too strong and often conflict.
We can fix the following points as the most important that such a controller must accomplish:

1) Ability to keep the process at the desired operating point.
2) Fast and smooth response to set point changes.
3) Keeping the transient behavior within certain ranges and avoiding excursion into dangerous zones.
4) Asymptotic stability.
5) Parametric insensitivity. (Resilience to plant changes).
6) Avoiding use of excessive control action (Ringing).

To these exhaustive constraints, we must add the usual characteristics of industrial chemical processes:lack of complete understanding of the phenomena, non-linearities and multiple steady states, interaction among variables, presence of dead time, high sensitivity of coupled variables etc. Authors such as Foss (30), Kestenbaum (48), Lee and

Weekman (56) suggest that the appropriate control system should be designed together with the process. But there are many processes that present a variable set of parameters or structure depending on the operating point and therefore, some periodic tuning must be done anyway.

The adaptive approach.-

The roots of the term adaptive control can be traced to the early 50's, but the first study on the subject by a control engineer was presented by Kalman (45), who designed and built an electronic analog device able to control a system while retuning its parameters.

The exact definition of adaptive control has been debated many times, and there exist different points of view depending on the users. To put an end to the semantic problem, Truxal states that an adaptive controller is one designed from an adaptive perspective. Later, Goodwin and Sin (37) specified as adaptive controllers all those algorithms that combine an on-line parameters estimation with a standard control design.

Interest in adaptive control resulted from the need to control plants that are poorly understood, with changing parameters and often with stochastics characteristics.

Due to many causes, mainly to hardware problems, Kalman's work went unnoticed and did not find echo in the academic world nor the process industry. The interest in

self-tuning systems was reinitiated by the work of Peterka (75) Astrom et al.(10) and, Astrom and Wittenmark (11), who applied recursive least square estimation and minimum variance strategy to develop their self-tuning regulator. This time, the advances in electronics and the new powerful algorithms, made possible the incorporation of the new methods to several industrial processes. Other schemes were introduced later, and there exists now a vast literature in adaptive control, where all kind of combinations between estimation algorithms and control system design are presented. The literature is mainly oriented to single input single output processes although a number of multivariable examples are considered. A valuable contribution to the field is the work of Goodwin and Sin (37) that provides a unified treatment for identification and control design, making easier the understanding of the many approaches to be found in literature. Seborg et al (80, 81) present a very complete and comprehensive survey, oriented to chemical engineering processing.

Even though adaptive methods represent a significant advance, some serious problems remain, such as the selection of weighting parameters, input/output pairing, decoupling algorithms, pole placement and, most important, stability considerations. So far, stability can be proved only for some specific linear systems.

A systematic approach:Internal Model Control.-

Garcia and Morari (32) refer to Frank as the originator of a novel approach called internal model control (IMC) that consists of an extension of the Smith's predictor. Joseph and Brosilow (44) used a similar idea for inferential control. Later, Garcia and Morari (31) developed the IMC system to its present state. The main idea is, through the "internal model", to open the loop behavior to proceed with the controller design as if it were feedforward control. The advantage of doing this is a very transparent insight into the stability conditions. At the same time, the controller design or better said, the difficulties in the design, are clearly related to the limitations of the plant itself.

The main drawback of the IMC design is that it requires a complete knowledge of the plant, with very precise models and ranges of possible variations.

The aim of this project is to unify the best characteristics of both schemes. Adaptive methods have a great capacity to deal with unknown or almost unknown plants but they present problems with the selection of some critical parameters very specific to each system. On the other hand, IMC structure gives clear hints about the limitations of the process and the best theoretical control that can be constructed. To accomplish this objective nevertheless, an exhaustive knowledge of the plant is necessary.

The most important problems to solve are:

-Identification of the multivariable plant.

-Controller design looking for the "plant inverse".

-Robustness of the eventual controller (Filter design.)

Expected results are:

An adaptive system that uses discrete operation, as seen from the computer side, and will be able to handle continuous or discrete processes, generating a stable control function following the structure of internal model control.

CHAPTER II

REVIEW

2.1.- SINGLE VARIABLE SYSTEMS

Lee and Weekman (56) state that the single most difficult problem to overcome while designing a system control is understanding the process itself. Now, chemical processes can in many ways pose different kinds of problems than those usually found in modern control theory. This is because they are characterized by a large number of inputs and outputs, large and varying transport delays, reciprocal responses (non-minimum phase), strong non-linear properties, a high degree of interaction. Furthermore, their parameters or even their structure can change from one operating point to another. Although some stochastic disturbances usually occur, the nature of chemical processes is rather deterministic.

The problems mentioned above have created an extensive interest in the design of control systems able to adapt or automatically adjust their controller settings in such a way as to compensate for the changing properties of the plant.

The search for adaptive controllers started about 30 years ago with the pioneering work by Kalman (45), but only in the last decade have some real applications been made, mainly based on the works of Wittenmark (90) and

Astrom and Wittenmark (11).

The successful application of the theory to practice was due to the development of faster and more reliable self adaptive algorithms and to the advances in electronics that made possible the implementation of these algorithms in small, fast, reliable and cheap microprocessors. Quoting Astrom (12), the typical characteristics for a computer in 1958 were 20 ms for a multiplication and mean time between failures (MTBF) of 50 to 100 hours. The figures were improved in the sixties to 100 microseconds per multiplication and the MTBF around 1000 hours. Today we can talk of multiplication time of 7 microseconds and MTBF of 20000 hours. Microcomputer prices have also dropped in such a way that computer control can now be considered as an alternative no matter how small the application.

Although there exist thousands of papers published in adaptive control with all kinds of combinations of estimators and control algorithms, we can distinguish between two general sets or categories. The first category assumes that the plant changes cannot be measured or anticipated. The second category assumes the process changes can be measured or at least inferred from other measured variables. In this case, a table approach could be used and in some way, a gain Kc could be updated any time some change has occurred in the plant. This scheme is known as gain schedule and in principle can maintain an adequate gain margin of stability in

spite of plant variations. Nevertheless, the gain schedule method takes for granted a complete understanding and modeling of the plant. In this work, in the section dealing with non-linear systems, this concept was extended through the adaptive calculation of a matrix gain using only primary information about the process such as Arrhenius's term in a chemical reaction.

With regard to the first set or category, there are hundreds of proposed algorithms but, surprisingly, they can roughly be classified into two general types : explicit and implicit methods. In the explicit approach a model is assumed and the parameters are estimated. Before calculating the control variables, some manipulation with the model is necessary. This is also called the indirect method because the estimated parameters do not directly appear in the control law. In the implicit form, a predictive model is employed and the parameters are also used as the control law coefficients (that is, the control parameters are directly updated).

Regarding the control law algorithm, there is a wide variety of techniques but, according to Seborg et al (81) they can be classified, somewhat arbitrarily and just for the sake of simplicity, in the following four types:

1.- k-step ahead adaptive control

2.- Pole-zero assignment

3.- Model reference control

4.- Other approaches

1.- Predictor methods

In the work of Astrom and Wittenmark (11), the feedback control is designed to minimize a cost function J = variance(y) where y is the output of the system. This minimum variance law can be reinterpreted as an optimal k-step ahead predictor, with k the dead time of the process, and a controller designed to make this prediction equal to a desired value r. (Normally assumed zero for a regulator).

As a remark, this algorithm is one of the simplest because, after using the predictor model and estimating its parameters, the control signal can be easily calculated as a linear combination of present and past outputs and past inputs. The most important inconvenience of this method is the excessive control action that is required and the instability when dealing with non-minimum phase systems. Besides, the algorithm is very sensitive to model order, the transportation delay must be known and, it doesn't have a tuning parameter.

Wittenmark (90) improves the self-tuning regulator including some modifications that provide integral action, feed forward control and non zero setpoint tracking.

Clarke and Gawthrop (20) extend the work of Astrom and Wittenmark by including in the cost function penalty terms with all the important variables that participate in the

process: y, r and u.

$$J = E((Py(t+k) - Rr(t))^{**}2 + (Qu(t))^{**}2)$$

where P, Q, R are appropriate weights, E is the expectation operator and r is the reference or setpoint. A less general form of J is:

$$J = E((y(t+k)-r(t))^{**}2 + \mu u^{**}2)$$

with μ as a tuning parameter.

This version of the self-tuning control is a notable improvement because the control can now deal with non-minimum phase systems and has a tuning parameter. The system is still sensitive to the model order and the dead time must be known in advance. To eliminate this requirement, Vogel's method (88)) can be employed making implicit an eventual dead time by increasing the number of parameters in the polynomial denominator of the transfer function. (Called B using the estimation nomenclature).

$$B = b_1 z^{-1} + b_2 z^{-2} + \ldots + b_n z^{-n}$$

If the unknown real delay turns out to be k, then k leading coefficients b_1, b_2, ..., b_k are zero.

Theoretical results

Some very important issues associated with the self-tuning regulator (STR) and self-tuning controller (STC) such as convergence robustness and stability, have received a great deal of attention. The analysis of these properties is very complex and abstract since the differential equations describing the closed loop performance are non-linear and time variant. The global stability and convergence have been proved in some specific cases by Goodwin et al(35) and, Goodwin, Johnson and Sin (36), mostly for discrete systems with constant parameters.

To achieve industrial acceptance, the self-tuning controllers have to be robust;that is, their performance must hold during upsets in the plant, such as changing parameters, unexpected disturbances and hardware failures. Unfortunately, the available theoretical studies concerning robustness are scarce and restricted to specific conditions. (Goodwin et al. (36)).

2.- Pole assignment

Wellstead and Prager (89) present a new strategy based on placing the poles of the closed loop system in such a way as to shape the process dynamical behavior to the wish of the designer. The proposed method is able to cope with unstable, non-minimum phase plants and it is robust to time varying transportation delays. The main drawbacks are the

computation on-line of diophantine equations at each sampling time and closed loop poles selection.

In 1983 Allidina and Hughes (5) published a very general algorithm intended to cover all the possibilities. The proposed general adaptive scheme includes the minimum variance controller of Clarke and Gawthrop and encompasses the model reference adaptive control presented by Egardt (27). Though the scheme is developed for deterministic system, the authors claim that after some suitable changes, the same sequence of calculations can be used for stochastic systems. To proceed with the algorithm, several diophantine equations must be solved on-line. No trends to multivariable extensions are shown and most important, no example or application is shown.

Corripio and Tompkins (24) discuss commercial applications of adaptive controllers using pole placement.

3.- Model reference systems

The basic idea is to make the output of an unknown plant asymptotically track the output of a given reference model. The outputs are compared then the controller parameters are modified to reduce the difference to zero. The original idea is due to Whitaker and was further developed by Monopoli (64) and Landau (51). To help with the design, some stability considerations are introduced such as Lyapunov stability criterion and Popov hyperstability criterion.

Lindorf and Carrol (58), present a comprehensive survey of adaptive design using Lyapunov methods. Porter and Tatnall (75) extend the Liapunov method to multivariable systems.

The major inconveniences with the model reference approach are the lack of capability to deal with disturbances and in general, the lack of any methodology to generate the reference model. In spite of all the advances about stability and boundedness of input and output vectors, Rohrs (78) made evident the poor robustness properties of the adaptive model reference design when some hidden dynamic characteristics of the process have gone unnoticed by the model.

4.- Other approaches

Linear Quadratic Methods

Among others, we can quote the optimal linear quadratic gaussian (LQG) methods. A quadratic cost function is minimized over an infinite time horizon. Usually, the LQG method is applied to state space models, leading to Riccati equations that are difficult to solve. (These equations are non-linear, and they must be solved backward). To make the problem easier, the steady state approximation is often employed.

Grimble (38) applied the optimal LQG to a polynomial transfer function using implicit and explicit approaches. He claims the algorithm is able to handle non-minimum phase systems allowing at the same time the use of integral

action. Only single input single output, linear time invariant and discrete models are tried as examples.

Clarke, Kanjilal and Mohtadi (22, 23) extend the LQG method to plants with dead time. They use a canonical state space representation as a clever way to decrease the number of parameters to estimate. They also solve the Riccati equation iterating only once at each sampling time, starting each time from the covariance calculated at the last sampling time. All the examples presented by the authors are single input single output discrete, linear and time invariant models.

Long term predictive and extended horizon

Lee and Lee (55) present an adaptive control with long term predictor, appropriate to deal with non-minimum phase systems. They design a control function using the same technique as the k-step ahead predictor, but for a parametric dead time m, greater than or equal to the known delay, and a quadratic cost function. If the system is multivariable, m is chosen greater than or equal to the minimum delay in the row number i.

The main difficulties with the method are the number of restrictions imposed over the system.

Ydstie (94) presents a similar approach with the advantage that his extended horizon method is less sensitive to model order and doesn't need to know the transport delay. Furthermore, the time delay can be time variant.

Deterministic time variant systems

Important advances in proving convergence and stability for linear time invariant systems have been achieved by Goodwin et al (35, 36). These results were extended by Evans et al (28) to a class of non-linear systems. The convergence of the estimation algorithm for invariant parameters is a guarantee for the parameter error boundedness if the plant is varying slowly (Anderson and Johnson (7)).

Xianya and Evans (93) present a control procedure able to handle unknown systems with rapidly varying parameters. The convergence proof for linear systems is also provided. Nevertheless, the proposed algorithm requires doubling the number of parameters to be estimated.

Gomart and Caines (34) present very general proofs for stability and robustness for adaptive control of time varying systems. They work with continuous rather than discrete models. No examples are presented.

2.2.- MULTIVARIABLE SYSTEMS

Usually, a real process control device involves several loops that are often interacting. In addition, we must expect in this case that, unlike the single variable process, the transportation delay is not characterized anymore by a scalar because each input u(i) presents a particular dead time when associated with the output y(j). Another problem is that the transmission zeros (equivalent to the zeros of the scalar transfer function) may lie in an unstable region even though no direct expression between the output y(i) and inputs u(1),u(2),...,u(n) shows this. (The converse is also true that is, the presence of an unstable zero in one element of the matrix not necessarily implies a transmission zero).

Much of modern control theory concerns the design of controllers for perfectly known linear plants. Some of the problems such as decoupling, adequate selection of the controlled variable-manipulated variable pairs and closed loop poles placement may be elegantly solved by using modal control techniques, (Moore (65)), or internal model control, (Garcia and Morari (31)).

A second approach is to consider the plant unknown and extend the adaptive techniques to the multivariable case.

Borison (15) uses the single input-single output adaptive scheme applying it to multiple input multiple output

plants by assuming that the whole system may be considered as a number of single loop controllers. The advantage of this philosophy of using autonomous controllers is the easy extension to any number of inputs and outputs, but in this approach, we find exactly the same problems described for single variable processes.

Koivo (49) designs a control system based on a single step optimal control in such a way as to minimize the steady state input and output variances. The algorithm leads to a quadratic gaussian cost function and therefore a Riccati matrix equation must be solved at each sampling time. Other drawbacks of the technique are the restrictions on the process because it applies only to linear, discrete, time invariant systems with stable invertible zeros and known transportation delays.

Morris et al.(67) develop a general self-tuning controller that includes feedback and feed forward action. It is applicable to non-minimum phase plants with different delays in each loop and with multirate sampling. They present some interesting results including experimental work with a distillation column. The main problems arise with the selection of several weighting matrices and the solution at each sampling time of diophantine matricial equations. The transportation delays must be known a priori and in general, the algorithm is very complex.

McDermott and Mellichamp (62) describe a multivariable

self-tuning controller that handles non-minimum phase, unstable systems with time varying delays. The algorithm is able to decouple and provide for closed loop pole placement. The main inconvenience of the method is an overwhelming complexity. Decoupling is achieved approximately by solving a large number of algebraic equations at each sample together with an optimization problem in order to balance the number of equations and unknown variables. Besides, the algorithm used for automatic placement of the poles must be solved each time the controller is activated. No indications are given about the computer time. Nevertheless, it can be considered as one of the most complete algorithms ever published for multivariable adaptive systems.

Lang et al.(53) present a generalized self-tuning controller with decoupling properties. This time, the closed loop poles are specified by the designer and not calculated on line. The decoupling properties are attained through some algebraic artifices increasing the number of parameters by a factor of 3 or 4.

Chien et al. (19) extend the algorithm presented by McDermott improving the decoupling characteristics but all the limitations of McDermott's algorithm are also applicable to this extension.

Agarwal et al.(1, 2) present a non-linear self-tuning controller they claim their algorithm performs better than conventional controllers based on linear models.

2.3.- INTERNAL MODEL CONTROL

Inverse response and time delay are perhaps the most characteristic properties found in chemical plants. They are often associated with troubles in the control loop behavior.

For conventional control, dead time appears as a limitation to the gain and therefore to the control quality, due to stability considerations.

In 1957 Smith (82) developed a simple methodology to overcome the dead time problem and today the technique is called Smith's Predictor. In spite of its theoretical importance, Smith's predictor has been replaced in practice by Dahlin's controller (24) that can be considered as special case of pole-zero design procedure. Nevertheless, Dahlin's controller cannot deal with nonminimun-phase systems. The ringing effects can be reduced following the modifications introduced by Touchstone and Corripio (86). These authors also include the use of instrumental variable to reduce the bias in parameters estimation caused by measurement noise.

Ogunnaike (70) presents a different and attractive approach based on a change of variables. The disadvantage of both methods is that the plant (including dead time), must be known in advance. Smith's predictor is an early precursor of a new and powerful control structure known as internal model control. On the other hand, Ogunnaike's algorithm though presented for known systems, can be extended to

unknown plants.

Alevisakis and Seborg (3, 4) extend the Smith's predictor to multivariable plants. The algorithm is not flexible and doesn't allow different delays but only one value associated with all the outputs.

Ogunnaike and Ray (71) propose a multidelay compensator able to deal with different time delays associated with each variable. The philosophy of including a model of the plant in the controller design is also presented by Joseph and Brosilow (44), through an inferential control method.

Garcia and Morari (32) refer to Frank as the first researcher who pursued the systematic approach to the design of controllers using the plant model in parallel with the real plant but great credit is due to Garcia and Morari (31), who developed the previous ideas to the present state. Through several papers, they shape in a solid form almost all of what is known today about the novel control structure called internal model control. They present IMC for single input single output processes, studying its most relevant properties and its relationships with other control structures such as model algorithm control, dynamic matrix control etc. In 1985, Garcia and Morari (32, 33) extended the IMC design to multivariable plants. The problem related to the synthesis of a realizable and stable controller are studied in detail. In the same way, they present the filter design and the conditions under which a given model for a

given plant will provide a stable closed loop for some value of the filter parameter. Their presentation is accompanied by several simulations that show the improvements over conventional methods;at the same time, the transparency of the procedure is emphasized.

Economou et al (26) extend the internal model concept to the control of non-linear systems. They present in detail the conditions under which a non-linear system is invertible. Nevertheless, they did not succeed in constructing a satisfactory analytical inverse and therefore, they use numerical procedures. The most important drawbacks of the method are its complexity and the deep knowledge of the system that is necessary. The use of the Jacobian matrix and the large number of differential equations to be solved makes it difficul to apply to multivariable processes.

Robust IMC Design

Internal model structure provides information that allows the safe design of controllers for given nominal plants. The problem of robustness arises due to the fact that it is impossible to determine a model that represents adequately the plant under all operating condition. So far, a great deal of effort has been dedicated to the study of robustness, that is, to the stability of the closed loop in spite of plant-model mismatch.

Zames (96) presents a complete study about the condi-

tions for achieving stability when using feedback control. The input-output stability is guaranteed if the open loop gain is less than one. The result is general for non-linear systems and can be extended to multivariable processes provided a consistent definition of norm of a matrix is used.

Morari (66) makes a detailed analysis about resilience for linear plant under the IMC structure. First he shows the equivalence, from an algebraic point of view, between IMC and any other feedback scheme. As the analysis goes on, the simplicity and superior overview of the internal model method is put in evidence. Then he presents how some inherent plant characteristics determine its resilience. The main problems to deal with are the non invertible elements such as dead time and transmission zeros, constraints on the manipulated variables and mismatch between plant and model.

Holt and Morari (41) present a procedure to minimize the effects of dead time on plant dynamic resilience. The multiple delays case is studied following a methodology far superior to that of the interactor matrix developed by Wolovich and Falb (91). The advantages of total and partial decoupling are also discussed. Last but not least, the procedure can be easily implemented in a computer program.

Laughlin et al (54) develop a technique to design robust controllers. First they map all possible plant variations into an uncertainty complex region then, following IMC methodology, the robust controller is designed. A spe-

cial matrix transformation method is employed in the procedure. This method is also used by Kantor (46, 47) with the same purposes, but Kantor's development is for multi input multi output (MIMO) systems and far simpler without complex diagrams. It remains to be seen to what extent the variations in the plant models are eventually known.

Zafiriou and Morari (95) base the robustness of the controller on the design of a low pass filter using the singular value approach. Besides the singular value calculations, the uncertainty ranges of the model must be known and the filter parameter is calculated from equations dealing with generalized gradients.

Palazoglu and Arkun (72) design a robust tuning procedure based on the singular values and their sensitivities. The main limitations of the method are that it requires a complete knowledge of the system and the eventual modifications the process can experience.

Manousiouthakis and Arkun (60) study the robust controller design under the structure of internal model control using a hybrid algebraic-topological approach. (They call the structure the Model Reference Scheme). In the first part, they assume a perfect model and proceed to design the controller as a feed forward one, using the results about factorization obtained by Pernebo (73). The second part deals with imperfect model and therefore, a feed back signal exists. To establish the topology of the model, the range

of possible variations must be known. The method is also based on singular values, assumes linear systems with known properties at infinity and the mathematical procedure is rather complex.

Rotea and Marchetti (79) present a linear quadratic regulator under the frame of internal model control. They use a canonical state space model and a Riccati equation must be solved. They assume the process is linear and time invariant. Perfect knowledge of dead time and model parameters is required.

Arulalan and Deshpande (9) develop a new approach for controller design using IMC structure. Their method, called simplified model predictive control, is based on the premise that it is always possible to design a controller that yields a closed loop response to set point changes, at least as good as the open loop response. The principal appeal of the algorithm is its simplicity. The following remarks can be made: A model of the process must be known, the decoupling properties are unsatisfactory and finally, it is not appropriate for non-minimum phase processes.

Svoronos (83) introduces an adaptive internal model control for a single input single output, linear, time invariant, discrete system. The modified IMC scheme is intended to improve the response of the process to disturbance changes and it is specially appropriate for dealing with slow and unstable systems. The control is not applicable to

non-minimum phase systems. After an extensive review of control literature, Svoronos' approach seems to be the only paper where the adaptive procedures are applied together with IMC structure.

CHAPTER III

THEORY

Introduction

Most of modern control theory relies upon a good knowledge of the system to be controlled. Our aim is to reduce the requirements of process modeling to a minimum. Nevertheless, the usual ways of representing a system will be useful as a frame of reference.

There are several ways of describing the dynamical behavior of a process, and we will study briefly three of them that appear to be the most important.

1.- State Space Representation

The state of a system is defined as the minimum number of its properties that we must know to study its dynamical behavior. The general form is a set of differential (difference) equations relating these properties and their derivatives with external forcing variables, called inputs, that drive the system from a given operating point (state) to another.

Usually we write

$$\begin{aligned} dx(t)/dt &= f(x, u) \\ y &= g(x, u) \end{aligned} \qquad \begin{aligned} &x: \text{state vector} \\ &u: \text{input vector} \\ &y: \text{output vector} \end{aligned} \qquad 3.1$$

for a general class of systems.

Whenever f is a linear expression or provided the system can be approximated by a linear expansion, the set (3.1) becomes

$$\begin{aligned} dx(t)/dt &= Ax + Bu \\ y &= Cx + Du \end{aligned} \qquad 3.2$$

If A, B, C and D are constant matrices we call the system linear invariant. Introducing an adequate similarity transformation w = Px with P a non singular matrix related to the stability matrix, we see that equation (3.2) becomes

$$\begin{aligned} dw(t)/dt &= \bar{A}w + \bar{B}u \\ y &= \bar{C}w + Du \end{aligned} \qquad 3.2b$$

where $\bar{A}$ has a special form known as companion matrix with the minimum number of parameters needed to represent the system.

The solution for set (3.2) is given by

$$x(t1) = \exp(A(t1-to))x(to) + \int_{t_o}^{t_1} \exp(A(t1-v))Bu(v)dv \qquad 3.3$$

An important expression can be derived for sampled (discrete) systems, assuming the zero order hold (ZOH) hypothesis applies:

Making to = kh and t1 = (k+1)h where k is the sampling time and h is the sampling period, we have

$$x((k+1)h) = \exp(Ah)x(kh) + \int_{o}^{h} \exp(A(t1-v))Bu(kh)dv \qquad 3.4$$

or

$$\begin{aligned} x(k+1) &= \emptyset(h)x(k) + \Gamma(h)u(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \qquad 3.5$$

The state space is the most complete representation of a process and contains the maximum amount of information about it.

For a linear, time invariant system, the dynamical behavior is determined by the eigenvalues of the matrix A.

2.- Transfer Function Matrix representation

This is a mapping of the inputs into outputs relating the Laplace transform of the vector of outputs with the Laplace transform of the vector of inputs. Assuming constant matrices A, B, C and taking the Laplace transform of the expression (3.2) with zero initial state, we have:

$$(sI - A)x(s) = Bu(s)$$

$$y(s) = Cx(s) + Du(s)$$

or

$$y(s) = (C(sI - A)^{-1} B + D)u(s) = G(s)u(s) \quad 3.6$$

A similar form is derived for the discrete state space equation (3.5) by using the Z transform.

The transfer function matrix contains less information than the state space representation unless no cancellation has occurred. Only the observable and controllable parts of the system are described by the matrix G(s). In spite of lesser information, this method is widely used because the variables that appear are the properties that the operator

can measure and manipulate.

3.- Matrix Fraction Description

This is an extension of the single input single output transfer function and can be derived from the matrix G. Let d(s) be the monic least common denominator of the elements gij(s) of G, then:

$G(s) = N(s)/d(s)$ with N(s) polynomial matrix. We can also write

$$G(s) = N(s)^{-1}(d(s)I) = (d(s)I)^{-1} N(s) \qquad 3.7$$

and we speak of right and left representations. The matrix fraction is not unique and in general we have:

$$G(s) = NrDr^{-1} = Dl^{-1}Nl \qquad 3.8$$

When the greatest common divisor between (Nr, Dr) or (Nl, Dl)is a unimodular matrix, (Nr, Dr) or (Nl, Dl) are said to be coprime. Whenever degree of det(Dr) = degree of det(Dl) = m = order of the matrix A, no cancellation has occurred, (Nr, Dr) or (Nl, Dl) are coprime, the system is controllable and observable and, the roots of det(Dr) are the eigenvalues of the matrix A. (Chen (17)).

Poles and zeros of G(s)

A polynomial matrix G(s) can always be reduced to its

Smith-McMillan form through a sequence of unimodular multiplications. (Patel and Munro (72b)). If the number of inputs n is equal to the number of outputs m then G(s) is equivalent to

$$M(s) = \mathrm{diag}(a_i(s)/b_i(s)),$$

where ai divides all ai+k and bi divides all bi-k. We define the poles of the system as the roots of all bi(s) = 0 and the transmission zeros of the system as the roots of all ai(s) = 0. Poles and zeros play respectively an important role in the stability of the process and the multivariable system inverse.

3.1.- TOWARD IMC STRUCTURE

The existence of transportation delay in a process has attracted the interest of controller designers for many years. For conventional analog controllers, the bandwidth and the reset rate are limited to avoid instability with the closed loop system.

Smith (81) proposes a novel method to overcome the problems associated with dead time. Working in a clever way with the block diagrams he first improves the proportional band and reset rate by designing the controller as in a system without delay. Then he introduces a secondary loop associated to the controller. The transfer functions G1 and G2 that define the process (except the dead times) appear in this secondary loop in such a way that the final closed loop

characteristic equation doesn't contain the dead times. The final block diagram appears in Fig. 3.1

Though the Smith predictor is sensitive to modeling errors and is not easy to implement using conventional equipment, it is the first successful attempt to modify the controller employing the plant model explicitly.

Joseph and Brosilow (44) design a control system starting from a different point of view. Their objective is to infer some process properties from secondary measurements. Doing that, they introduce a model G# in parallel with the plant G to be controlled. The purpose of the model G# is to isolate the effects of the unmeasured disturbances on the process output.

3.2.- PROPERTIES OF IMC

Let's represent a controlled multivariable linear process by a conventional feedback block diagram as shown in Fig. 3.2.

$G \in Ra^{mxn}$ process transfer function matrix

$C \in Ra^{nxm}$ controller transfer function matrix

$y \in R^{m}$ output vector

$u \in R^{n}$ control vector

$d \in R^{m}$ disturbance vector

Ra is a commutative ring of rational stable functions with real coefficients.

Ra^{mxn} is a noncommutative ring whose elements are (mxn) matrices with elements belonging to Ra.

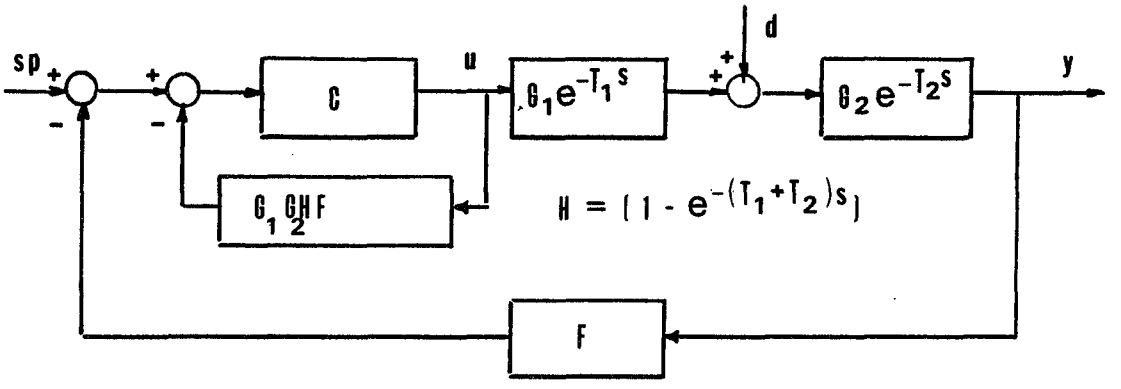


Fig. 3.1

Smith Predictor

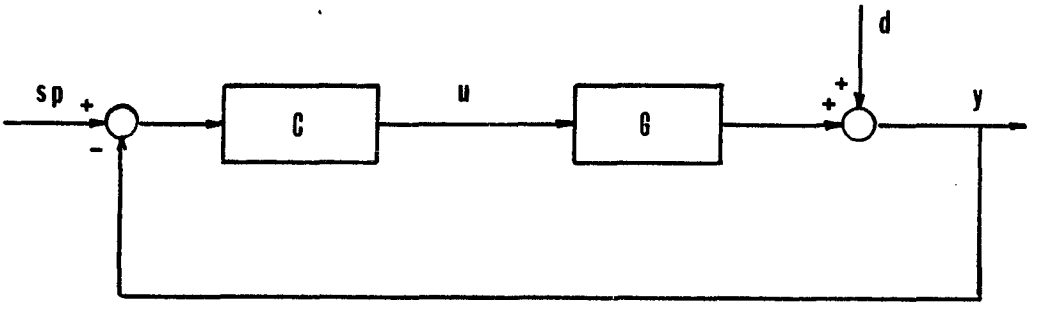


Fig. 3.2

Common Feedback Loop

R is the field of real numbers and R^n is the n-dimensional vector space defined on R.

The system matrices and vectors are implicitly functions of a general variable p that represents the Laplace variable s for continuous processes and the z variable if the system is discrete.

The output y is easily calculated as

$$y(p) = G(I + CG)^{-1} C(sp-d) + d \qquad 3.9$$

Defining perfect control as:

$$y(p) = sp(p) \quad \text{any d and any time;} \qquad 3.10$$

we can conclude that perfect feedback control is achievable if and only if $|C| \longrightarrow \infty$ (whatever definition of norm we use).

From the block diagram

$$u(p) = C(I + GC)^{-1} (sp-d) \qquad 3.11$$

and

$$Gu = GC(I + GC)^{-1} (sp-d) \qquad 3.12$$

for perfect control Gu = y-d = sp-d

therefore

$$GC(I + GC)^{-1} = I \qquad 3.13$$

and this requires that the right inverse of G exits. The matrix G has a right inverse if and only if rank of G = m so,

the number of manipulated variables n must be at least equal to the number of outputs m. The Moore Penrose generalized inverse must be understood if n > m.

The configurations developed by Smith and Joseph and Brosilow, were adopted by Garcia and Morari (31), under the structure shown in Fig. 3.3.

The model G# appears in parallel with the plant and with the controller. After replacing the dashed block by its algebraic equivalent, we obtain the final internal model control structure as shown in Fig. 3.4.

The following equivalences hold:

$$Gc = (I + CG\#)^{-1}C \qquad \text{or}$$

$$C = Gc(I - G\#Gc)^{-1} \qquad 3.14$$

No specific form is required on C except the conformability of the matrix products. Furthermore, the block diagram in Fig. 3.4 is algebraically equivalent to the conventional feedback control in Fig. 3.1.

After some manipulations, the output y is expressed by:

$$y(p) = G(I + Gc(G - G\#))^{-1}Gc(sp-d) + d \qquad 3.15$$

From this, the well known properties of IMC structure follows. (Garcia and Morari (31)).

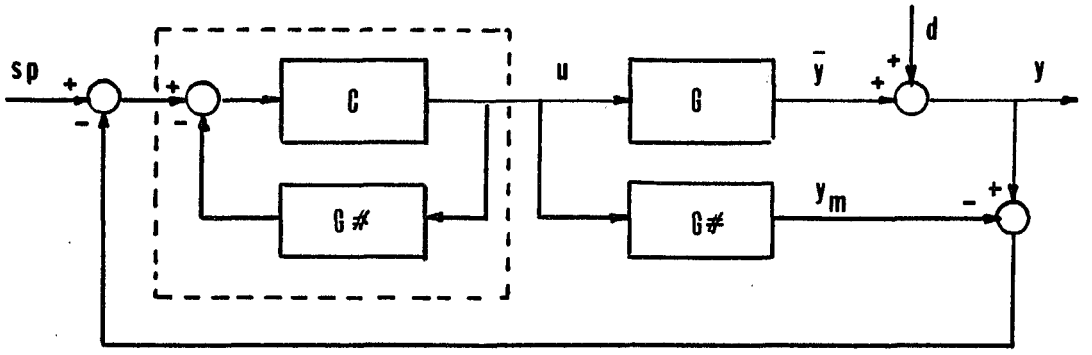


Fig. 3.3

Parallel Model Structure

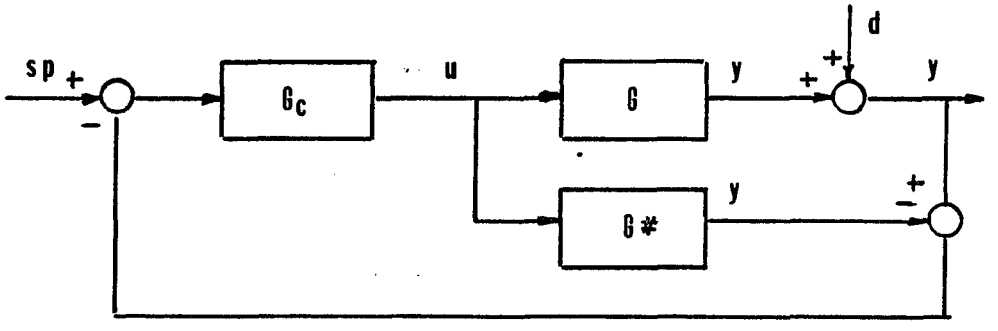


Fig. 3.4

Internal Model Control

1.- Perfect modeling;if G# = G

$$y = GGc(sp-d) + d \qquad 3.16$$

then, the stability of the closed loop will be assured if G is open loop stable and Gc is chosen also stable.

2.- Perfect control;If Gc is chosen as the right inverse of G in (3.16) then perfect control is achieved. (At least in theory). The need for the right inverse of G follows naturally in this approach and is equivalent to equations (3.13) and (3.14).

3.- Asymptotic perfect control;zero offset is asymptotically obtained if Gc is chosen such that:

$$\lim_{p \to p^*} Gc(p) = (G\#(p^*))^{-1} \qquad 3.17$$

where

$$p^* = \begin{cases} 0 & \text{if } p = s, \text{ continuous systems} \\ 1 & \text{if } p = z, \text{ discrete systems} \end{cases}$$

Making the controller Gc equal to G inverse is seldom possible. Several problems arise related to inherent properties of the plant itself.

1.- Non causal inverse. Whenever some delay is present in the plant, its inverse leads to non realizable functions.

2.- Nonstable inverse. If G has transmission zeros in a non-

desirable zone (lambda zone), then its inverse will be unstable. The lambda zone is defined by extension of the conventional RHP subset of the s-plane, or, for discrete systems,as the region outside the circle $|z| = 1$. The designer usually defines the lambda zone according to some particular criterion. (For example, to avoid oscillation).

3.- Improper inverse. Even when dealing with causal and stable inverses, strictly proper process models will lead to improper inverses that present extreme sensitivity to high frequency perturbations.

The construction of the control function must consider all these problems. Usually, the transfer function $G\#$ is represented in the form of a product of two factors:

$$G\# = G_+ G_- \qquad 3.18$$

where G_+ contains all the non invertible properties of $G\#$ and G_- is called the lambda invertible part of $G\#$. (The plus is adopted by analogy to the unstable positive semiplane s. In the book of Astrom (12) for example, the converse is assumed).

The controller is then expressed as:

$$Gc = (G\#)^{-1} G_+ F \qquad 3.19$$

The term F corresponds to a filter function that compensates

the improper form G_-^{-1} .

G_+ is not unique and several techniques have been developed to construct it.

Under perfect modeling, the output y(p) becomes:

$$y = G(G\#)^{-1} G_+F(sp-d) + d \quad \text{or}$$

$$y = G_+F(sp-d) + d \qquad 3.20$$

The difficulties associated with the plant structure appear explicitly represented by the factor G_+. For example, if the plant has a delay, exp(-Ts), the setpoint tracking or the disturbance rejection can be achieved only after a time T has elapsed, these features are included in the term G_+ in (3.20).

3.3.- ROBUST DESIGN

The preceding discussion about the controller design was done assuming perfect modeling of the plant. Under uncertain models, equation (3.15) can be modified introducing explicitly the filter F. The first condition for stability follows immediately:

$$\det (I + GcF(G-G\#)) \neq 0 \qquad 3.21$$

for any value of the generalized variable p and for any G#

that can represent the plant.

Manousiouthaki et al (60) define a topology over the set of all possible variations of G with :
$E(G, \delta) = (G' \varepsilon Ra^{mxn}(p)$ s.t. $g'ij \; E(gij, \delta ij) \; \forall \; i=1, 2, \ldots, m, \; \forall \; j=1, 2, \ldots, n)$ and a similar topology is defined for the scalar transfer function gij
$E(gij, \delta ij) = (g' \varepsilon Ra(p)$ s.t. $|g'ij-gij|po < \delta ij(w) \; \forall \; po \varepsilon \delta \Lambda)$.
$\delta \Lambda$ is the boundary of Λ and $\delta ij(w) = \max|gij(po)-g'ij(po)|$, $\forall \; po \varepsilon \delta \Lambda$.

A norm definition for matrices is necessary. The spectral norm, based on the maximum singular value of the matrix, is adequate for this purpose. (See Arkun et al (8)).

To achieve asymptotic perfect tracking, the closed loop transfer matrix must accomplish:

$$G(p^*)Gc(p^*)(I+F(p^*)(G(p^*)-G\#(p^*))Gc(p^*))^{-1}F(p^*) = I \qquad 3.22$$

applying (3.17) and after some manipulations

$$F(p^*) = I \quad \text{for any plant variation} \qquad 3.23$$

The stability condition requires that the roots of the expression det $(I+F(G-G\#)Gc)$ must never encircle p^* and therefore, the determinant must never change sign as the plant varies over $E(G, \delta)$. When G is equal to the nominal plant $G\#$, the determinant becomes one. This fact fixes the

sign for all the allowable variations of G and therefore :

$$\det (I+F(G-G\#)Gc) > 0 \qquad 3.24$$

for any G over E(G,) and any p.

If $p = p^*$, (3.17), (3.23) and (3.24) imply :

$$\det (G(p^*)\ G\#(p^*)^{-1}) > 0 \qquad 3.25$$

so, the changes in the plant G with respect to a nominal model G# cannot be such that restriction (3.25) is broken.

Levien and Morari (57) present a distillation column model and show how the change of three parameters in a (3x3) system leads to instability. (Though the changes are rather excessive :100%).

Disturbance rejection and Filter Design

The control vector is found directly from Fig.3.2:

$$u = Gc(sp-d) \qquad 3.26$$

If G# can be factorized into a lambda invertible matrix G_- and a non invertible part G_+ such that $|G_+| = 1$ then Gc is constructed as:

$$Gc = G_-^{-1} F = (G\#)^{-1} G_+ F \qquad 3.27$$

Therefore, replacing the last expression into (3.26)

$$u = G^{-1} F(sp-d) = (G\#)^{-1} G_+ F(sp-d)$$

we see that the norm of u is given by:

$$|u| = |(G\#)^{-1}||sp-d| < |u|max \quad \text{and}$$

$$|sp-d| < |G\#||u|max$$

For a regulator problem, the magnitude of the disturbance is bounded by the norm of G#. This norm can be represented by the greatest singular value of G#

$$|sp-d| < \sigma_M (G\#)|u|max \qquad 3.28$$

At high frequency, σ_M (G#) is small and the magnitude of the allowable disturbance is also small due to the saturation of the control vector u.

Zafiriou and Morari (95) use the singular value approach to design an optimal filter. They recommend a simple diagonal structure for F to keep the number of variables small at the optimization stage. Nevertheless, the algorithm requires the calculation of gradients and it is rather complex.

Kantor (46, 447 using the spectral radius instead of the singular values, proposes another filter design method under internal model control structure.

CHAPTER IV

MODELS AND ESTIMATION

4.1.-MODELS

A model is a representation of a reality that we pretend to know. There are many levels of modeling, several ways to proceed with them and a given phenomenon usually has associated with it more than one model.

In a first level, we can consider some physical scaled representation of the actual system. For example, architects build small scale models of houses and with them they are able to study the best orientation of the house with respect to hours of light, relationship with the landscape etc.

In a second level, more advanced applications are sought and our purpose is to use the model to describe relationships among the different properties of the process. The set of equations that describes such relationships is called the mathematical model.

A model may change and normally the modifications are suggested by the knowledge we get from the previous ideas. A beautiful example of model evolution is given by the structure of the hydrogen atom. First, it is thought of as a solid sphere. Some gas kinetic properties are studied using this primeval model but, it is impossible to predict from it other properties of matter. The next approximation considers

the atom as a nucleus with an orbiting electron. More information is gained but some phenomena are still left without explanation. A third stage considers the electron orbit as a probability density and a dual wave-particle behavior for the electron. Again, more knowledge is obtained. Of course, we pay for the advances with more complex mathematics, so it is necessary to establish a trade off between investment and return.

Generally, the starting point for building a model is a set of relationships known as conservation principles.

Mathematically we talk of static or dynamical models, continuous or discrete , deterministic or stochastic etc.

We must never pretend that a model is the real life actual system. In fact, our models have a more pragmatic sense and their acceptance is guided by their usefulness, that is, we are normally more concerned whether the model can fit experimental data than with philosophical aspects of the representation of reality.

Sometimes, specially when testing a given identification procedure, we compare the results with those obtained by simulation of the "true" plant. Of course, this expression must be understood in a figurative sense since the "true" plant is nothing more than a higher level model used to generate some data such as a real plant would provide.

Input-output models.

The properties associated with a system are called variables and are classified as inputs and state variables. Those properties that exist independent of the system itself are called inputs. The properties that are a consequence of the inputs and the modifications introduced by the plant are called state variables. Some of these state variables or combinations of them that have an external manifestation are called outputs. From the mathematical point of view, all the inputs are equivalent but, according to engineering characteristics, such as our ability to handle them, they are classified into perturbations and control variables. In this way, the steam flow to a process and the environment temperature are both inputs, but it is much more comfortable to think of the steam as an operative variable and to treat the temperature as a perturbation over which no possible control action can be exerted.

Computer oriented model

A fundamental problem is how to describe a continuous system connected to a digital computer. The discrete operation of the computer requires that the continuous flow of information from the plant be converted to a sequence of numbers that the computer can process. The signals are sent to the computer through an analog-to digital converter and returned to the plant with a digital-to-analog device.

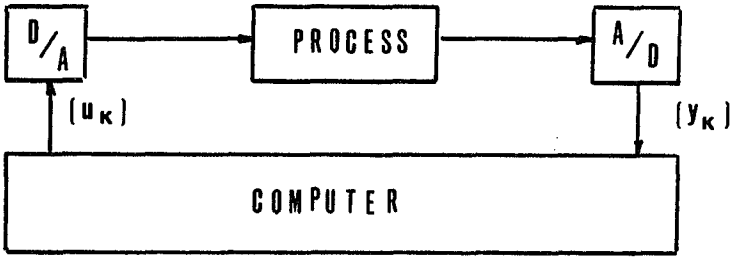


Fig.4.1

Digital Control

The symbols (u(k)), (y(k)) mean sequences of numbers corresponding to the values of u(t) and y(t) at the sampling time k Δt. An internal clock decides when the sampling must be done.

4.2.-IDENTIFICATION

System identification is the construction of models starting from input-output data for a given plant. Identification is important to theory but it also plays a valuable role in the design of control systems and can be considered as the experimental building of models.

The identification stage is an open problem with several degrees of freedom. This freedom can be bounded according to our knowledge of the process to be identified. The black box approach assumes we know nothing about the system, but of course this is a very restrictive way. Usu-

ally, the previous knowledge about the process is significant, and we have approximate ideas of its performance with different kinds of stimuli, relative magnitude of response, dead time in the response, presence of non-linear effects like Arrhenius's expression etc.

The complete identification problem requires the determination of a structure and the related parameters. The task is made easier if we assume a model structure (for example, a transfer function matrix), and we just calculate the associated parameters. In that case, we have reduced the identification problem to one of parameter estimation.

Astrom (12) points out the following features to help with identification:

Experimental planning
Selection of model characteristics
Estimation
Validation

4.2.1.-Experiment design

To design an experiment leading to the identification of a plant, we need to solve several questions such as what and when to measure, how many measurements, what kind of inputs etc. Many of these questions will have an appropriate answer only after the plant has been identified. As usual, some design decisions must be taken without sufficient in-

formation. Of course, previous knowledge of the plant and skilled operators will be a worthy help. On the other hand, restrictions do exist;some of the variables are very difficult or impossible to measure (catalytic activity for example), some instruments are too expensive others too slow etc. Furthermore, too much information can be confusing and difficult to process.

With respect to inputs, our main concern should be to reduce the perturbations to the minimum amount. If possible, the normal operating values must be used. Whenever additional inputs are required, the excitation must be persistent, that is, rich in information. Appropriate inputs for identification purposes are pseudo random binary sequences (PRBS) and white noise signals. The amplitude of the signals must be selected specifically for each system. In the same way, the period of the binary sequence must be chosen according to the dominant time constant. To improve the estimation, it is desirable to use normalized input-output pairs.

Sampling period.

The selection of the sampling period h is of paramount importance; and unfortunately, linked with the unknown dynamics of the system. It would appear, as a first guess, that h should be chosen very small to have a better representation of continuous signals, but this would mean a great increase in the computer load. Even worse, if the process is

slow, a large sampling frequency means an ill conditioned information matrix with the subsequent inversion problems. Astrom (12), recommends the choice of 1/h six to ten times the bandwidth of the closed loop or two to three per rise time. Sometimes, multirate sampling is introduced to avoid the use of complicated antialiasing filters. This is specially convenient for multivariable systems where one of the variables is slow relative to the others. If the sampling periods are synchronized and h1 = nh2 with n integer, then the ordinary theory for discrete systems can be applied.

4.2.2.- Model selection

The choice of the model is rather subjective and perhaps the best way is to build it while thinking of its applications. For control purposes, this leads to a simple structured form known as ARX that uses the sequences of numbers generated through the sampling procedure.:

$$y(t)+A_1 y(t-1)+ \ldots+A_{N_{AM}} y(t-N_{AM}) = B_1 u(t-1)+\ldots+B_{N_{BM}} u(t-N_{BM}) + e(t) \qquad 4.1$$

$A_1,\ldots,A_{N_{AM}}$, $B_1,\ldots,B_{N_{BM}}$ are (mxm) matrices with unknown parametric elements, N_{AM} and N_{BM} are the model order parameters and e(t) represents a white noise uncertainty associated with measurements. The name ARX means autoregressive model with exogenous variable. (Sometimes, the names ARMA or

DARMA are also used).

After introducing the shifting operator q such that

$$q^{-1}f(t) = f(t-1) \qquad 4.2$$

equation (4.1) becomes

$$(I+A_1q^{-1}+\ldots+A_{N_{AM}}q^{-N_{AM}})y = (B_1q^{-1}+\ldots+B_{N_{BM}}q^{-N_{BM}})u + e \qquad 4.3$$

We define the polynomials matrices:

$$A(q) = I + A_1q^{-1}+\ldots+ A_{N_{AM}}q^{-N_{AM}}$$
$$B(q) = B_1q^{-1}+\ldots+ B_{N_{BM}}q^{-N_{BM}}$$

then equation (4.3) becomes

$$A(q)y = B(q)u + e \qquad 4.4$$

solving for y we introduce the transfer matrices:

$$y(k) = A^{-1}Bu(k) + A^{-1}e(k) = G\#(\theta, q)u(k)+ He(k)$$

According to the system complexity (and our knowledge of it), more sophisticated expressions, such as non-linear effects, can be introduced. Once the structure of the system has been adopted, the order N_{AM} and N_{BM} must be chosen. Ljung

(59) propose several methods to test the order, but most of them are highly complex using spectral analysis of G# or the rank of the information matrix. A more pragmatic approach consists in the comparison of performances for two models with a fresh data set (cross validation). Another practical way is the minimization of the loss function with respect to the order. Some care must be taken to avoid overfit and therefore overparametrization.

4.2.3.- Estimation algorithm

Our main interest is directed to adaptive control and therefore, to on-line, recursive algorithms. Goodwin and Sin (37) present complete information about the available methods.

For use on-line, we need a simple and fast algorithm and the recursive least square seems to accomplish these requirements. Note that recursive least square is intended for linear estimation but, the linearity is referred to the parameters and not to the system itself. Agarwal and Seborg (1, 2) present a good example of a non-linear adaptive control for single and multiple variable systems.

We now introduce the matrix of parameters Θ given by

$$\Theta = (\theta_1, \theta_2, \ldots, \theta_M) = (A_1, \ldots, A_{N_{AM}}, B_1, \ldots, B_{N_{BM}})^T \quad 4.5$$

where Θi are NPx1 vectors, Ai and Bi are mxm matrices and NP $= N_{AM} + N_{BM}$.

At each step N, the vector $\Theta i(t)$ is determined as function of the last estimation $\Theta i(t-1)$ and a gain vector K(t) in such a way to minimize the loss function:

$$J(\Theta) = \left(\sum_{1}^{N} e_K^T e_K \right)/N$$

$$e_K = y_K - \hat{y}_K \qquad 4.6$$

Upon defining the regressor $\emptyset$ as :

$$\emptyset^T = (y^T(t-1),\ldots,y^T(t-N_{AM}),u^T(t-1),\ldots,u^T(t-N_{BM})) \qquad 4.7$$

the following recursive algorithm results:

$$\Theta i(t) = \Theta i(t-1) + K(t-1)(yi(t)-\emptyset^T(t-1)\Theta i(t-1)) \qquad 4.8$$

$$K(t-1) = P(t-1)\emptyset(t-1)(1 + \emptyset^T(t-1)P(t-1)\emptyset(t-1))^{-1} \qquad 4.9$$

$$P(t) = P(t-1)-K(t-1)(1+\emptyset^T(t-1)P(t-1)\emptyset(t-1))K^T(t-1) \qquad 4.10$$

This multivariable estimation algorithm follows Borison's presentation (15) and the proof of minimization can be found in Goodwin and Sin (37).

Some modifications can be done to improve the character-

istics of the algorithm. To cope with time varying systems, it is necessary to eliminate or at least to attenuate the weight of the old data. This is done by introducing a forgetting factor in the loss function:

$$J(\theta) = \sum_{1}^{N} \lambda^{N-k} (y(k) - \theta(k)\phi(k))^{**}2/N$$

The value of λ is less than one and appears in the updating of P and K.

$$K(t-1) = P(t-1)\phi(t-1)(\lambda + \phi^{T}(t-1)P(t-1)\phi(t-1)) \qquad 4.11$$

$$P(t) = (I - K(t-1)\phi^{T}(t-1))P(t-1)/\lambda \qquad 4.12$$

with initial estimation

$$\theta(0) = 0, \quad P(0) = \alpha I \; ; \quad \alpha > 10^{6}$$

The forgetting factor presents some problems when the system has been running steadily and, the input data are no longer persistently exciting and P grows exponentially. If some sudden change occurs under this condition, the estimation algorithm is highly sensitive and the controlled system becomes unstable unless some detuning is applied. This phenomenon has been studied by several authors and one of the best approaches is due to Fortescue (29). Based on the

information content of the data, he develops a procedure to make the forgetting factor variable, with a value very near one when the estimation has converged. If some change is detected, the forgetting factor decreases and the estimator is reactivated.

Astrom (12) proposes a very simple algorithm to achieve a variable forgetting factor:

$$\lambda = 1 - \mu e^2 / \bar{e}^2$$

where e is the prediction error, $\bar{e}$ is the mean value over a given period of estimation and μ is a constant factor

We use $\bar{e}^2 = (1+\phi^T(t-1)P(t-1)\phi(t-1))$ (Goodwin and Sin)

then $\lambda = 1 - \mu e^2/(1+\phi^T P\phi)$ which is basically Fortescue's algorithm.

Covariance matrix factorization

From a numerical point of view, the information matrix P can be very ill-conditioned specially for large systems. The convergence of the estimation is related to the condition number γ defined by:

$\gamma = \rho\max\ P(\theta)^{-1} / \rho\min\ P(\theta)^{-1}$ with ρ eigenvalue of P so that if γ is closer to one the convergence is improved and

$$|\theta(t)-\theta o| \quad < \quad |\theta(0)-\theta o|$$

where θ is the current estimation matrix, θo is the true matrix of parameters and $\theta(0)$ is the initial estimation matrix. To improve the performance, Bierman and Thornton (14), factorize P as the product of three matrices:

$P = U\ D\ U^T$ where D is a diagonal matrix and U is upper triangular. Another approach is to consider the Cholesky decomposition or any other method that allows to express P as the product QQ^T . The loss function is not affected by the transformation but, the ratio between the largest and smallest eigenvalues of Q is the square root of that of P and therefore, the new system will be better conditioned than the original. Peterka (76) presents an algorithm called REFIL written in such a way that it can be directly implemented in a computer program. This program, with the variable forgetting factor introduced by Fortescue was the workhorse in all our estimations

Remarks.

1.- No information in advance is required for the delays of the system. It is advisable however to make a rough estimation to fix N_{BM} .

2.- The same algorithm REFIL can be used for extended least square or non-linear systems by changing the way in which the regressor vector ϕ is constructed.

4.2.4.- Model validation

Once the model structure has been fixed, the order has been selected and the parameters calculated, the natural questions that arise are:

Does the model fit the data?

Is it simple enough for control purposes?

How easy (or not) is it for the computer to estimate the parameters?

There are also other questions concerning the philosophical approaches to the true plant, but from a practical point of view they are less relevant than the former.

For linear systems a well known practice to evaluate different models is through the comparison of Bode plots, but this approach is not appropriate for multivariable processes.

With a fixed structure, only two degrees of freedom are left: N_{AM} and N_{BM}. To answer those questions, we looked for the minimum values for these parameters that allowed a reasonable fit. The practice was to start with $N_{AM} = N_{BM} = 1$, increasing N_{BM} we looked for the minimum of the loss function

$$J = \sum_{1}^{Ns} (y-y\#)^2/Ns.$$

Then, N_{AM} was incremented by one and the procedure was repeated.

The cross validation resulted in a very good tool that is self-implemented in a process with internal model control. The crucial part of IMC is to work in an open loop form, that is, with perfect modeling. First the model was estimated using some persistent excitation and then, the fresh data was entered through changes in the reference vector. The difference (G-G#)u is then, the best index of how well the model behaves.

CHAPTER V

FACTORIZATION AND INVERSION

Introduction

For linear systems perfect internal model control requires the inversion of the plant model represented by the transfer function matrix G#. As discussed in chapter three, this inversion is seldom possible and therefore, an approximate inverse is sought after the matrix G# has been factored into a lambda-invertible part G- and a non invertible term G_+.

We are concerned with rational proper functions G# but, due to the nature of the ARX model explained in chapter four and to the estimation algorithm, the matrix G# is already expressed as the product of two polynomial matrices:

$$G\# = A^{-1}(z^{-1})B(z^{-1}) \qquad 5.1$$

where A and B are defined in (4.3) and correspond to the partitioned matrix Θ defined in (4.5).

$$\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_M) = (A_1, \ldots, A_{N_{AM}}, B_1, \ldots, B_{N_{BM}})^T \qquad 5.2$$

From (5.1):

$$(G\#)^{-1} = B^{-1}(z^{-1})A(z^{-1}) \qquad 5.3$$

Expression (5.3) must be considered as a formal one because we still do not know under what conditions the inverse of matrix B exists.

Hereafter, the argument z corresponding to discrete systems will be used since it is the natural way for iterative estimation. Nevertheless, the development that follows is general and can be applied to continuous systems.

Expression (5.3) points out that the factorization and inversion problem is reduced from working with rational matrices to handling polynomial matrices. Therefore, the best linear control under IMC structure will be defined as long as we are able to obtain the factoring and inversion of the polynomial matrix B.

5.1.- MATRIX FACTORIZATION

By factoring a polynomial matrix B(z) we must understand the search for two new polynomial matrices $B_+(z)$ and $B_-(z)$ such that they have their zeroes in disjoint sets of the complex plane. Normally the Λ region (region of B_+ zeroes) is defined as the set of all points in the extended plane C+, such that these points have given properties. As a first step, lambda contains the unstable poles such as the points inside the unit circle, but it can be enlarged to other points that the designer does not want as poles. By exten-

sion we call a matrix lambda-stable if its poles are outside the lambda zone, though not necessarily all points inside lambda are unstable. Such factorization is always possible under very mild conditions (e.g., the matrix must be analytic) and is closely related to the minimal realization problem.(Bart et al.(13)).

5.1.1.- Transmission zeroes and poles of a matrix

The definition of the lambda-region is based on the stability properties of linear systems and therefore, on the concept of zeroes and poles of multivariable systems.

For a given matrix G with elements g_{ij} , scalar rational transfer functions, we define the transmission zero ρ as a value of z such that it reduces the column rank of G. That is, if ρ is a transmission zero of G, there exists a non zero vector b such that:

$$G(\rho)b = 0$$

(b is called a latent vector of G)

The definition of pole is not so expeditious and therefore we must first define the Smith-McMillan form of G.
Let d be the monic least common denominator of all g_{ij} , then:

$$G(z^{-1}) = N(z^{-1})/d(z^{-1}) \qquad 5.4$$

where $N(z^{-1})$ is a polynomial matrix.

we define:

a) $fo = 1$

b) fi = the monic largest common factor of all (ixi) minors of $N(z^{-1})$.

c) $\psi i = fi/fi\text{-}1$

Then, the diagonal matrix S

$$S = \text{diag}(\psi i) \qquad 5.5$$

is called the Smith form of the matrix N and can be obtained from N through a finite number of transformations.

Finally, dividing by $d(z^{-1})$ we have:

$$M(z^{-1}) = S/d(z^{-1}) = \text{diag}(ai(z^{-1})/bi(z^{-1})) \qquad 5.6$$

M is called the Smith-McMillan form of G. From M, the poles of G are defined as the roots of the polynomials $bi(z^{-1})$. Thus, the Λ-zone is defined as the collection of points in the complex plane that the designer doesn't want as roots of $bi(z^{-1})$. For example, for a discrete system we may define lambda such as:

$(z \varepsilon \Lambda \; / \; z \text{ s.t. } |z| > 0.8)$

Note that the Λ-zone includes stable points.

Once the Λ-zone is defined and the Smith-McMillan form of G has been obtained, the factorization of any matrix is rather trivial. The difficulty is hidden in the search for the Smith form. Although this is an elementary procedure, the numerical methods are unsatisfactory.

Kontos (50), uses an alternative form based on the elementary factors of the polynomial matrix and on its latent vectors. Unfortunately, the determination of latent vectors requires the singular value decomposition of the matrix for each of its transmission zeroes and therefore the method is not appropriate for on-line implementation.

5.2.- MATRIX INVERSION AND FACTORIZATION

So far, no available factorization algorithm seems to be appropriate for on-line calculation.

The following scheme was developed for obtaining an algorithm that reduces the matrix factorization to that of a scalar polynomial and that is appropriate for working on-line.

From (5.6) we can see that the zeroes of the matrix G, that will be the poles of an eventual inverse matrix, appear in the diagonal of the Smith-McMillan form of G and therefore their product is contained in the determinant of G. This leads to solving first the inversion problem. The factorization will then result as a consequence of the inversion procedure.

5.2.2.- Computer inversion of a square polynomial matrix

The inversion of polynomial matrices often arises in the analysis and design of control systems. For example, given a linear time invariant system represented by the state space equations, the determination of the transfer function matrix requires the inversion of the matrix sI-A. Also in the design of multivariable control , the inverse of G is required.

The inversion of rational matrices and specifically of polynomial matrices has been studied by many authors. One of the oldest methods is due to Leverrier as quoted by Chen (17). This method is considered of theoretical importance but not recommended for large matrices because it requires too many operations and tends to be unstable. The main purpose of Leverrier was to find the characteristic polynomial of a constant matrix and in this sense the method was improved by Frame (1949) and Faddeev and Sominskii (1949) as quoted by Householder (42). Later, it was modified by Inouye (43) and applied to the inversion of a more general polynomial matrix and not only to the elementary pencil defined by $\rho I - B$.

Inversion of the pencil $\rho I - B$

Given an (mxm) matrix B with constant elements $b_{ij} \in R$, the expression $\rho I - B$, where ρ is a non specified complex parameter and I is the (mxm) identity matrix, is called a

pencil of matrices.

It can be shown that:

$$(\rho I-B)^{-1} = (R_0\rho^{m-1} + R_1\rho^{m-2} + \ldots + R_{m-2}\rho + R_{m-1})/\Delta(\rho)$$

where $\Delta(\rho) = \det(\rho I-B) = \rho^m + a_1\rho^{m-1} + \ldots + a_m$

with a_i and R_i constant scalars and matrices respectively and defined by:

$$\begin{aligned} a_1 &= -\mathrm{tr}BR_0/1 & R_0 &= I \\ a_2 &= -\mathrm{tr}BR_1/2 & R_1 &= BR_0 + a_1 I \\ a_3 &= -\mathrm{tr}BR_2/3 & R_2 &= BR_1 + a_2 I \\ &\ldots & &\ldots \\ a_m &= -\mathrm{tr}BR_{m-1}/m & 0 &= BR_{m-1} + a_m I \end{aligned}$$

where tr stands for trace of the matrix. In other words, the algorithm allows us to express the inverse of the elementary pencil as:

$$(\rho I-B)^{-1} = \mathrm{adj}(\rho I-B)/\Delta(\rho) \qquad 5.7$$

providing an iterative method to compute a_i and R_i .

After using the fact that the inverse of any (mxm) monic polynomial matrix B with degree n

$$B = B_0 + B_1\rho + \ldots + I\rho^n \qquad 5.8$$

can be expressed in term of an elementary pencil of matrices, Inouye (43) modified the method of Leverrier producing a fast and reliable algorithm that reduces the inversion to handling (mxm) matrices.

Two problems remain :

1)The method is not completely general because the leading matrix B_n must be the identity matrix or at least non singular.

2)The algorithm is valid only with column (row) reduced matrices, that is deg det(B) = $\Sigma\ \delta_{ci}\ B(\rho)$

where δ_{ci} B is the highest power of ρ in all the elements of column i. Now, any non reduced matrix can be reduced through the multiplication of unimodular matrices but the condition is not easy to test and the search for the unimodular matrices is complex and not appropriate for on-line computation. Buslowicz (16) established an algorithm that improves Inouye's method making it less restrictive and faster.

Buslowicz algorithm:

Given $B(\rho) = B_0 + B_1\ \rho + \ldots + B_n\rho^n$; Bi (mxm);then: 5.9

$$\det B(\rho) = a_0 + a_1\rho + \ldots + a_r\rho^r \qquad 5.10$$

with r less than or equal to mn

Therefore:

$$B(\rho)^{-1} = \sum_{0}^{nm-n} Qk\rho^k/\det(B) = Q/\det(B) = adj(B)/\det(B) \qquad 5.11$$

with

$$Qk = (-1)^{m+1}\, Rm-1\ ,k \qquad ; k = 0,1,\ \ldots,n(m-1)$$

$$ak = ((-1)^{m+1} tr\ Gm,k\ )/m \qquad ; k = 0,1,..,mn$$

$$Ro,\ k = \begin{cases} I & \text{if } k = 0 \\ 0 & \text{if } k \neq 0 \end{cases}$$

$$Rj,k = Gj,k + aj,k*I \qquad ; j = 1,2,\ \ldots,m-1$$

$$Rm,k = Gm,k + am,k*I \qquad ; k = 0,1,\ \ldots,mn$$

$$Gi,k = BoRi-\ ,k+B1Ri-\ ,k-1+\ \ldots\ + BnRi-\ ,k-n$$

$$ai,k = (-tr\ Gi,k)/i \qquad ; i = 1,2,\ \ldots,m$$

set 5.12

Even though this algorithm is more complex than that of Inouye, it was implemented as a computer program called POLIN (POLynomial INverse) using FORTRAN 77.(See appendix B). The program worked efficiently under test conditions and also on-line integrated to the IMC controller. In the tests it showed that it is fast, reliable and free of numerical problems even when B is a very ill-conditioned matrix, because the adjoint of B always exists independent of the values of det(B).

Note that in the algorithm the augmented matrices have been replaced by an increased number of (mxm) matrices.

Factorization of the inverse of B

Our conjecture is that all the required dynamical information is contained in det(B). At this point we return to the variable z^{-1} specific for discrete systems. All the in-

version steps are valid if we replace ρ by z^{-1}. Solving numerically for det(B) = 0, det(B) can be separated into three dynamical factors and a constant.

$$\det(B) = a_0 + a_1 z^{-} + \ldots + a_N z^{-N} = z^{-ND}\,\Delta^{+}(z^{-1})\Delta^{-}(z^{-1})a_N \quad 5.13$$

ND represents the non-causal factor due to dead time, $\Delta^{+}(z)$ contains the unstable part of B (roots $\varepsilon\ \Lambda$), $\Delta^{-}(z^{-1})$ is the lambda stable factor of B (roots outside Λ) and, a_N is the coefficient of the highest power of z^{-1} in the determinant polynomial. This last factor is taken out to produce monic polynomials Δ^{+} and Δ^{-}.

We define the following parameters according to the factorization:

ND : global dead time of the process

N^{+} : number of Λ-roots of det(B)

N^{-} : number of stable roots of det(B)

N : total number of roots of det(B)

The above parameters have an internal meaning and were calculated at each iteration.

Let's illustrate the procedure with an example.

Assume $\Delta(z^{-1}) = 6z^{-3} + 39.8z^{-4} + 74.4z^{-5} + 58.4z^{-6}$

$$20.4z^{-7} + 2.6z^{-8}$$

such as would be provided by the inversion program.

Then

$$\Delta = z^{-3}(z^{-1}+.2394)(z^{-1}-1)(z^{-1}+1.61)(z^{-1}-2)(z^{-1}-3)2.6$$

with ND = 3, $N^+ = 2$, $N^- = 3$, N = 5 and

$$\Delta^+ = (z^{-1}+.2394)(z^{-1}-1)$$
$$\Delta^- = (z^{-1}+1.61)(z^{-1}-2)(z^{-1}-3)$$

In this example, Λ was assigned as the region outside the circle $|z| = 1$.

With the information provided by the factorization of the scalar polynomial Δ we are able to construct B_+, the non invertible part of B. To discard the non-causal part, a rough factor z^{-ND} I could be included, but this procedure leads to unnecessary time delays in the control action and therefore in the response of the system. Holt and Morari (41) show a better way to deal with the non-causal term, and we shall concentrate on developing numerical aspects of their algorithm.

Let βij be the elements of inverse B and

p_{ij} = min delay in the denominator of βij

q_{ij} = min delay in the numerator of βij

Compute

$$di = \max_i(\max_j(0, p_{ij} - q_{ij})) \qquad 5.14$$

then construct the matrix

$$D = \text{diag}(z^{-d_i})$$

The matrix D deals with the non-causal terms of B in a decoupled way. This decoupling is optimum in the sense that D provides the minimum amount of delay to each row in such a way as to cancel out the non-causal term z^{-ND} . If the most relevant dead times are not in the diagonal of B, the cancellation is not optimal but the decoupling characteristics are maintained.

According to the way in which B^- is calculated, adjB is a new polynomial matrix therefore, $p_{ij} = ND$ for all i, j and we only have to compute q_{ij} and thus the calculation of di is almost inmediate.

Finally B_+ was constructed in the following form:

$$B_+ = \frac{\Delta^+(z^{-1})}{\Delta^+(1)} \text{diag}(z^{-d_i}) \qquad 5.15$$

Therefore, the stable and causal inverse of B is :

$$B_-^{-1} = B^{-1} B_+ = \frac{\text{adj}(B)}{z^{-ND} \ \Delta^-(z^{-1})\Delta^+(1) \ a_N} \text{diag}(z^{-d_i}) \qquad 5.16$$

or

$$B_-^{-1} = \frac{QN}{\Delta^-(z^-) \ \Delta^+(1) \ a_N} \qquad 5.17$$

where QN is the matrix that results after multiplying adj(B) by matrix D and cancelling z^{-ND} .

The factor $\Delta^+(1)$ is included in (5.15) so that $B_+(1) = I$ and therefore $B_=^{-1}(1)$ becomes exactly $B^{-1}(1)$ when time goes to infinity.

5.3.- SYNTHESIS OF Gc.

Once the approximate inverse of B was constructed, the implementation of the control algorithm followed from expression (5.3).

$$Gc = (G\#)^{-1} F \cong B_=^{-1} AF \qquad 5.18$$

F is a diagonal matrix filter included to make the control action robust. For simplicity's sake, the filter was defined as a first order lag with elements $f_{ii} = (1- \alpha_i)/(1- \alpha_i z^{-1})$ with α_i tunable parameters.

Introducing expression (5.17) into expression (5.18)

$$Gc(z) = \frac{QN \; A \; F}{\Delta^-(z^{-1})\,\Delta^+(1)\,a} = \frac{H \; F}{\Delta^-(z^{-1})\,\Delta^+(1)\,a_N} \qquad 5.19$$

The control vector was calculated as:

$$u = Gc(sp-\bar{d}) = Gc\,\varepsilon = \frac{H \; F \; \varepsilon}{\Delta^-(z^{-1})\,\Delta^+(1)\,a_N} \qquad 5.20$$

The difference $sp-\bar{d} = \varepsilon$ is the generalized error in the IMC structure.

Making $F\varepsilon = x$, $Hx = v$, we have:

$$u(k) = (v(k)/(a_N \Delta^+(1)) - \sum_{2}^{N-} b_i\, u(k-i+1))/b_1 \qquad 5.21$$

with b_i coefficients of the stable polynomial $\Delta^-(z^{-1})$.
Thus, u(k) was generated as a function of present and past generalized errors and past inputs.

We can summarize this chapter by stating that a method for inverting and factoring a polynomial matrix B was successfully developed. The algorithm is based on the properties of the Smith-McMillan form of B but without searching for it. The method starts by inverting the matrix and then the matrix factorization is reduced to a scalar problem appropriate to be solved on line.

A complete example of the procedure is presented below

The following discrete model was used for representing a continuous distillation column:

$$N_{AM} = 1, \quad N_{BM} = 3, \quad m = 2$$

At arbitrary time = 64, the estimated parameters are:

$$A = \begin{bmatrix} 1. & 0. \\ 0. & 1. \end{bmatrix} + \begin{bmatrix} -.921 & -.022 \\ -.223 & -.747 \end{bmatrix} z^{-1}$$

$$B = \begin{bmatrix} .717 & -.016 \\ -.353 & .517 \end{bmatrix} z^{-1} + \begin{bmatrix} .005 & 0.004 \\ 1.076 & -1.017 \end{bmatrix} z^{-2} + \begin{bmatrix} -.005 & -.856 \\ -.734 & -1.235 \end{bmatrix} z^{-3}$$

The matrix B is then submitted to the inversion algorithm POLIN. The following information is returned:

$$\det(B)=(.3653-.7088z^{-1} -1.2109z^{-2} +.9253z^{-3} -.6262z^{-4})z^{-2}$$

$$\text{adj}B= \begin{bmatrix}.517 & .016\\ .353 & .717\end{bmatrix}z^{-1} + \begin{bmatrix}-1.017 & -0.004\\ -1.076 & .005\end{bmatrix}z^{-2} + \begin{bmatrix}-1.235 & .856\\ .734 & -.005\end{bmatrix}z^{-3}$$

Det(B) is then factored into a stable factor and a non-causal unstable part using Mullers method plus an ordering algorithm.

Therefore:

$$z^{-2}\Delta^{+} = (z^{-1} -.3494)(z^{-1} +.6305)z^{-2}$$

$$\Delta^{-} = (z^{-2}-1.7587z^{-1}+2.7203)(-.6262)$$

with ND = 2

This factorization is specific for ($z \in \Lambda$/ s.t. $|z| < 1$). If a different definition of Λ is used some of the roots will move from Δ^{-} into Δ^{+} .

The terms q_{ij} are calculated from the elements βij of the matrix adj(B):

$$\beta ij = \beta^{0}ij + \beta^{1}ijz^{-1} + \ldots + \beta^{r}ijz^{-r}$$

and

q_{ij} = K with K the first term such that

$$|\beta^{K} ij| > 10^{-n} * \max_{l}(|\beta^{l} ij|) \qquad 5.22$$

With this criterion we are neglecting parameters that are 10^{-n} times less than the maximum coefficient. A value for n

equal 2 or 3 was found appropriate.

In the example, one element of adj(B) is:

$$\beta_{11} = .517z^{-1} - 1.017z^{-2} - 1.235z^{-3} \text{ and therefore}$$

$$\max_{l}(|\beta^{l}_{11}|) = 1.235$$

for n = 3, β^{1}_{11} fits the criterion and so K = 1 ==> q11 = 1

From this d1 and d2 are calculated:

$$d1 = \max(\max(0, ND-q11), \max(0, ND-q21)) \text{ or}$$

$$d1 = \max(\max(0, 2 - 1), \max(0, 2 - 1)) = 1$$

$$d2 = \max(\max(0, 2 - 1), \max(0, 2 - 1)) = 1$$

The factor B_+ is calculated as:

$$B_+ = \frac{(z^{-1} - .3494)(z^{-1} + .6305)}{1.0608} \begin{bmatrix} z^{-1} & 0 \\ 0 & z^{-1} \end{bmatrix}$$

Finally the stable factor B_-^{-1} is:

$$B_-^{-1} = adj(B)*B \ /det(B) = QN/(\Delta^{-}(z^{-1})a_N)\Delta^{+}(1)$$

In this example since d1 = d2 , QN is equal to the matrix adj(B) multiplied by the variable z. In general, d1 ≠ d2 and a more involved calculation must be done (algorithm BMIN in appendix B).

CHAPTER VI

SIMULATION RESULTS

6.1.-SIMULATION

Simulation can be defined as the prediction of the system behavior by using models. It is a powerful and indispensable tool for testing control systems, nevertheless we must recall that it is impossible to cover through it all combinations of parameters, controllers and disturbances and therefore, analysis must be used as complement.

There are several computer packages oriented to simulation, with different characteristics according to the user's goals. Among these programs we can quote Statistical Analysis System (SAS) that includes some estimation features;Continuous System Modeling Program (CSMP) a complete model oriented package useful for solving differential and discrete equations;and LINPACK (LINear PACKage), a program designed for working with numerical matrices and appropriate for calculating inverse matrices, eigenvalues and singular value decomposition. More specific to control applications we may quote ORACLS and SIMNON (an interactive control program) but with restricted availability,

All these programs have in common the large amount of computer memory that they demand and the lack of flexibility

for working in an interactive way (except perhaps, SIMNON).

Our goal made necessary the implementation of a small package that is self contained and eventually could be run in computers with rather modest characteristics.

The following scheme depicted in Fig. 6.1 was adopted:

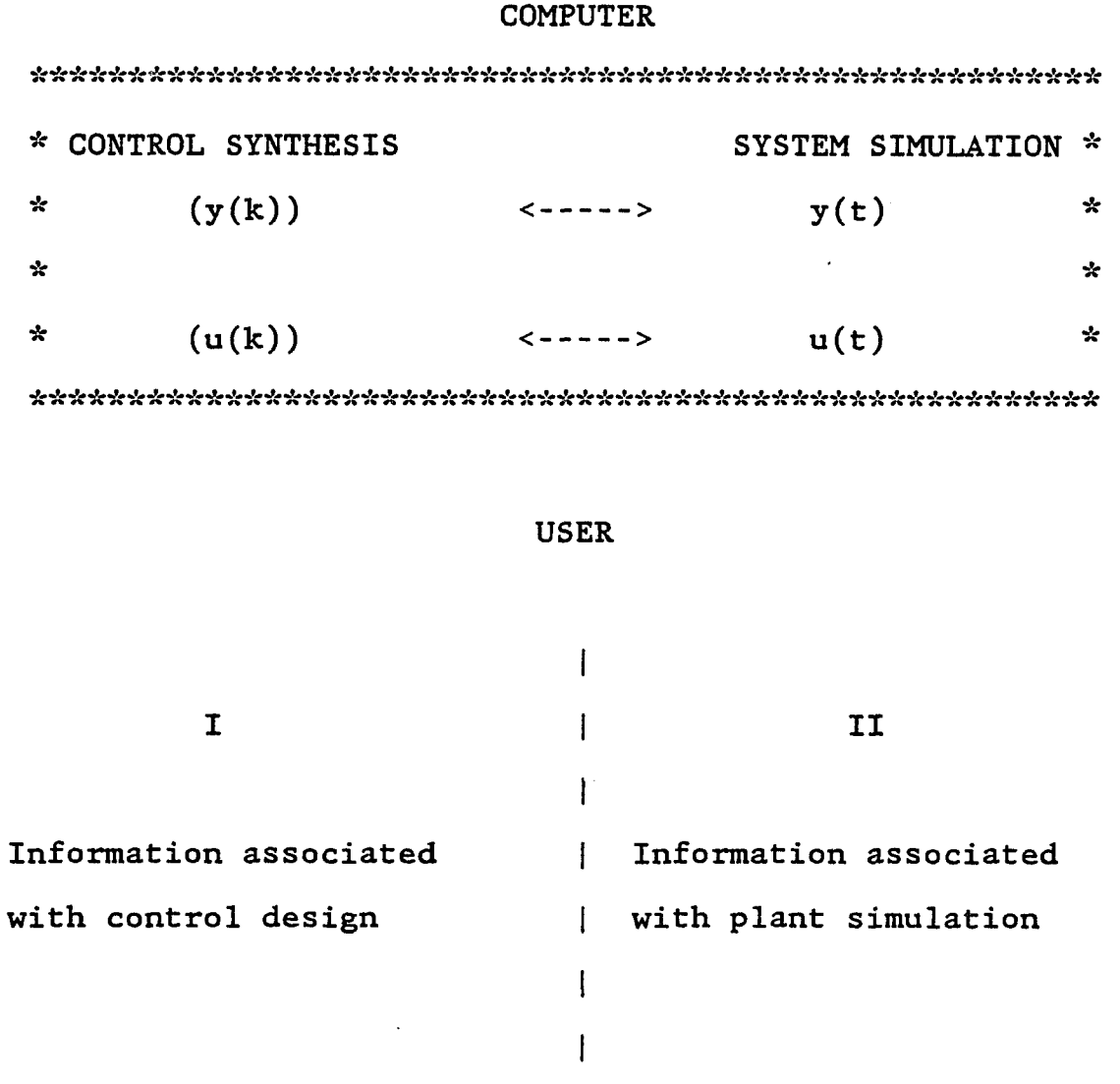


Fig. 6.1

In the first group we find:

Number of inputs and outputs m

Order of the model N_{AM} and N_{BM}

Sample period h

Initial set of estimated parameters $\theta(0)$

Stability zone (Λ-zone)

Gain schedule (for non-linear systems)

With the second group we must provide:

Number of inputs and outputs m

Order of the plant N_A and N_B (for discrete systems)

Differential equations and integration step (for continuous systems)

Initial conditions

Set point and load changes

Set of inputs for estimation purposes (u(k))

Characteristics of noise associated with measurements $N(x,\sigma)$

Open loop identification time

Total time of simulation (open loop + closed loop)

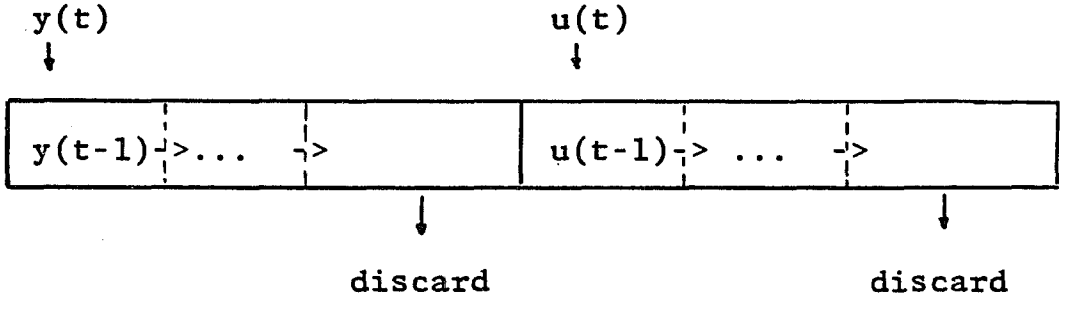
The only common information between the simulated process and the control synthesis was the number of inputs and outputs m, the open loop estimation time and present value of vectors y and u.

The simulation package was implemented using structured FORTRAN 77, but only a small number of its features were used such as instructions IF()THEN/ENDIF and DO 100 K = 1, (M**2+M)/2 for instance. These instructions are easy to change if a lower level of FORTRAN is available. Other more

powerful characteristics such as arrays with negative indexes were not used and therefore the shifted variable y(t-i) is represented by the FORTRAN variable Y(I).

Simulation structure

Once the simulation started, the information about the system was retained at each sample in a fixed length matrix through a shifting subroutine:



Then the regressor vector with variable length, according to the postulated model, was constructed:

$$\phi^T = (y^T(t-1),\ldots,y^T(t-N_{AM}),u^T(t-1),\ldots,u^T(t-N_{BM}))$$

where the supra index T means transpose.

The regressor vector and the estimation error vector defined by the product of the matrix of parameters Θ and the regressor vector ϕ, were sent to the estimation algorithm REFIL. The return was the set of polynomial matrices A(z) and B(z). Matrix B was inverted and factored and then the

control matrix Gc was computed. Finally, the control vector u(k) was computed according to equation (5.21).

6.2.- OPEN LOOP ESTIMATION

In this section we present some of the results obtained with the estimation algorithm without using any control scheme.

Some parameters must be specified in a heuristic way:

a) sampling frequency

b) magnitude of input

c) order of postulated model

d) simulation time

If the system is unknown, several open loop runs are necessary. However, this type of process seldom occurs and some information is always available. For instance, transient duration is a good indicator for fixing the sample interval and the total simulation time. In the same way, the nature of the process gives some clues about the magnitude of the inputs. For estimation purposes, some variable scaling may be appropriate. The estimation algorithm is more accurate working with inputs and outputs of the same order of magnitude so it is better for instance to work with T/To and C/Co than directly with T and C.

The order of the model has special importance when dealing with continuous systems with dead time. Our aim was to keep the order as small as possible reducing in this way

the number of parameters, but sometimes a large dead time makes necessary a large value N_{BM} to allow the best possible representation. The zero order hold (ZOH) hypothesis was enforced by keeping the vector u(k) constant during the intersample time.

Examples

6.2.1.- Discrete system. Paper machine headbox

The following model is presented by Borison (15)

$$y(t) + A_1 y(t-1) = B_1 u(t-1)$$

with

$$A_1 = \begin{bmatrix} -0.9901 & 0.008805 \\ -0.8061 & -0.770890 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0.8990 & -0.004590 \\ 19.3900 & 0.880520 \end{bmatrix} \qquad 6.1$$

We define the above expression as system B1

where

y1 = stock level

y2 = total pressure

u1 = stock input flow

u2 = air input flow

We define system B2 with:

a(1,1) is reduced in magnitude by 40%

b(2,1) is reduced in magnitude by 10%

The following parameters were used:

Simulated plant:Discrete system B2 for 25<t<61 and 81<t<100

Discrete system B1 any other time

Sampling period h: 1 second

Input vector u:PRBS with amplitudes (0.1, 0.2)

Initial information matrix $P(0) = 10^7 I$

Initial estimation $\theta(0) = 0$

Model:The simplest model with $N_{AM} = 1$, $N_{BM} = 1$ was used. The forgetting factor is initially set to one and changes when the estimation error is large. When the parameters converge, between time 0 and 25 sec., the forgetting factor remains constant near to one. At time 25 sec. two parameters are decreased in magnitude and therefore the estimation error increases modifying the forgetting factor and making the estimation algorithm active. After a while the new parameters have been calculated and the factor returns to one. The cycle is repeated each time the parameters change.

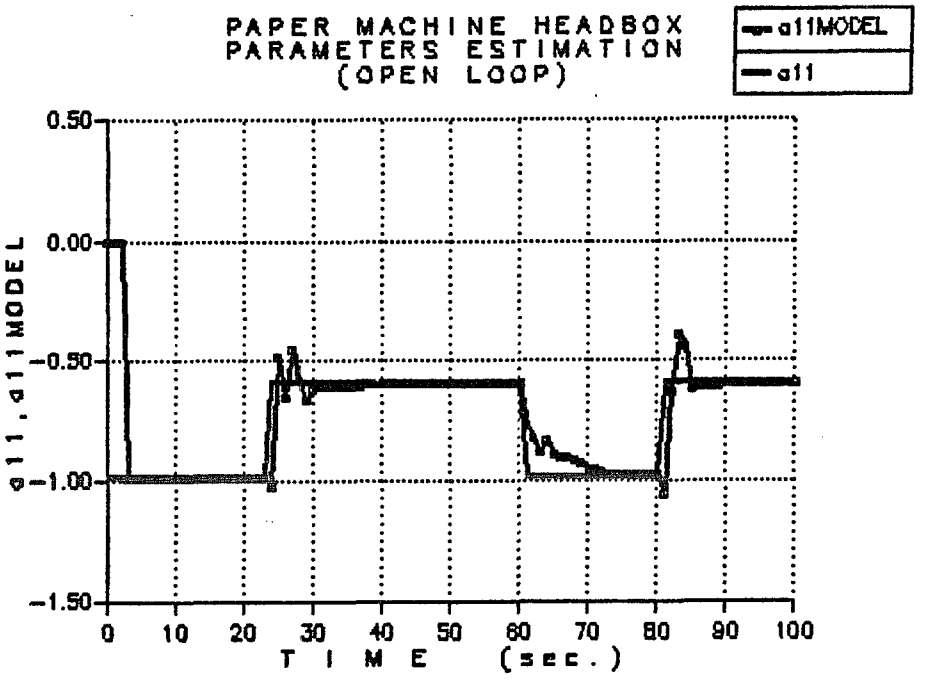
A very effective adaptation to the new values can be observed in Fig. 6.2.

6.2.2.- Continuous double effect evaporator model

A continuous model for a double effect evaporator is presented by Newell and Fisher (68):

$$\dot{x}(1) = -0.00156x(2)-0.1711x(3)-0.143u(2)+0.2174F1$$

$$\dot{x}(2) = -0.14190x(2)+0.1711x(3)-0.74F1+0.1434CF1$$


PAPER MACHINE HEADBOX
PARAMETERS ESTIMATION
(OPEN LOOP)
a11MODEL
a11
0.50
0.00
-0.50
-1.00
-1.50
a11, a11MODEL
0
10
20
30
40
50
60
70
80
90
100
TIME (sec.)

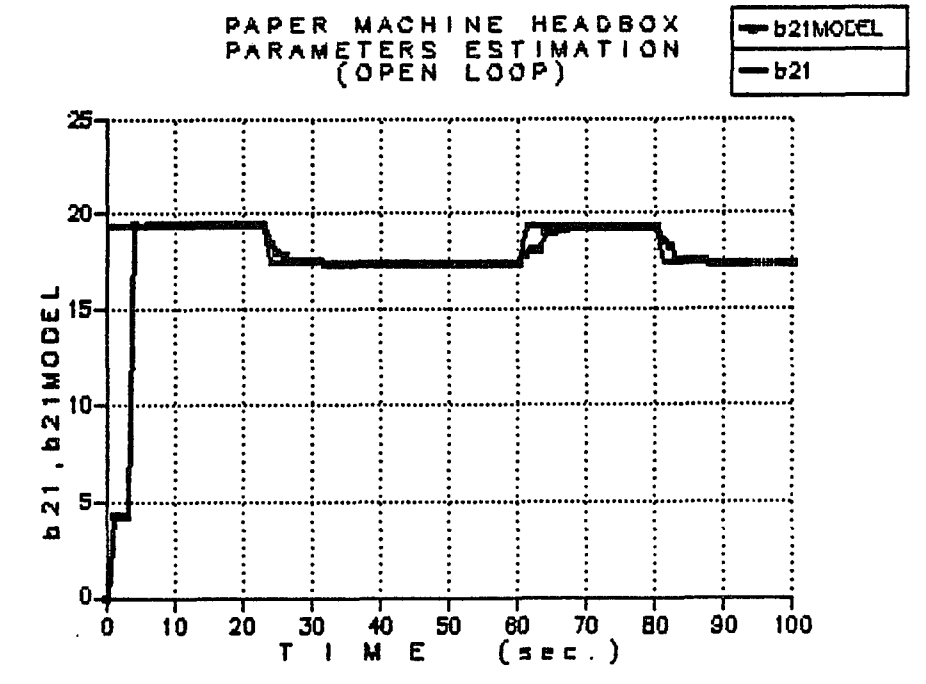

PAPER MACHINE HEADBOX
PARAMETERS ESTIMATION
(OPEN LOOP)
b21MODEL
b21
25
20
15
10
5
0
b21, b21MODEL
0
10
20
30
40
50
60
70
80
90
100
TIME (sec.)

Fig. G.2

$$\dot{x}(3) = -0.00886x(2)-1.1020x(3)+0.392u(1)-0.036F1+0.1814hF1$$

$$\dot{x}(4) = -0.00128x(2)-0.1489x(3)-0.00013x(5)+0.108u(2)-0.0592u(3)$$

$$\dot{x}(5) = 0.06050x(2)+0.1489x(3)-0.05910x(5)-0.486u(2)$$

set 6.2

The state variables are holdup, concentration and enthalpy in first effect, and holdup and concentration in second effect respectively.

The inputs to the system are steam flow S, bottom streams B1 and B2 or u(1), u(2) and u(3) respectively. F1, CF1 and hF1 are feed flow, feed concentration and feed enthalpy (perturbations). The measured outputs are defined by:

$$\begin{bmatrix} y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix}$$

Simulated plant: Continuous and fixed system represented by equation 6.2. Runge Kutta integration with fixed step of 0.125 min.

Sampling period h: 1 min.

Input vector u: PRBS with amplitudes (0.43, 0.79, 0.25)

Initial information matrix $P(0) = 10^7 I$

Window parameter $\sigma_o = 0.5$ (6 to 15 samples)

Initial estimation $\theta(0) = 0$

Measurement noise w(0.025, 0.10, 0.05), zero mean

Model:

$$A(z) = I + A\,z^{-1} + A\,z^{-2} \quad ;N_{AM} = 2$$

$$B(z) = B\,z^{-1} + B\,z^{-2} \quad ;N_{BM} = 2$$

The number of parameters to be estimated is given by

$$NPM = mxm(N_{AM} + N_{BM}) = 36$$

Fig. 6.3, 6.4 and 6.5 show the outputs y1, y2 and y3 together with the model outputs ym1, ym2 and ym3 respectively. From the figures, we can observe that the model output is hard to distinguish from the plant output after the parameters have converged. Finally for this example, Fig. 6.6 shows the evolution of two representative parameters.

6.2.3.-Continuous distillation column

Wood and Berry (92) developed a dynamical model of a pilot distillation column designed to separate a mixture of methanol and water. This model, or slight modifications of it, has been used for control testing purposes, among others, by Ogunnaike and Ray (71), McDermott and Mellichamp (61) and, Arulalan and Deshpande (9).

The following set of differential equations was used in this work to represent the continuous plant behavior:

$$\dot{x}(1) = -0.0600x(1)+(0.7680-0.4012Ind)u1(t-1)$$

$$\dot{x}(2) = -0.0476x(2)+(0.9000-0.5000Ind)u2(t-3+Ind)$$

$$\dot{x}(3) = -0.0917x(3)+(0.6055-0.2844Ind)u1(t-7+3Ind)$$

$$\dot{x}(4) = -0.0694x(4)+(1.3472-0.7361Ind)u2(t-3+2Ind)$$

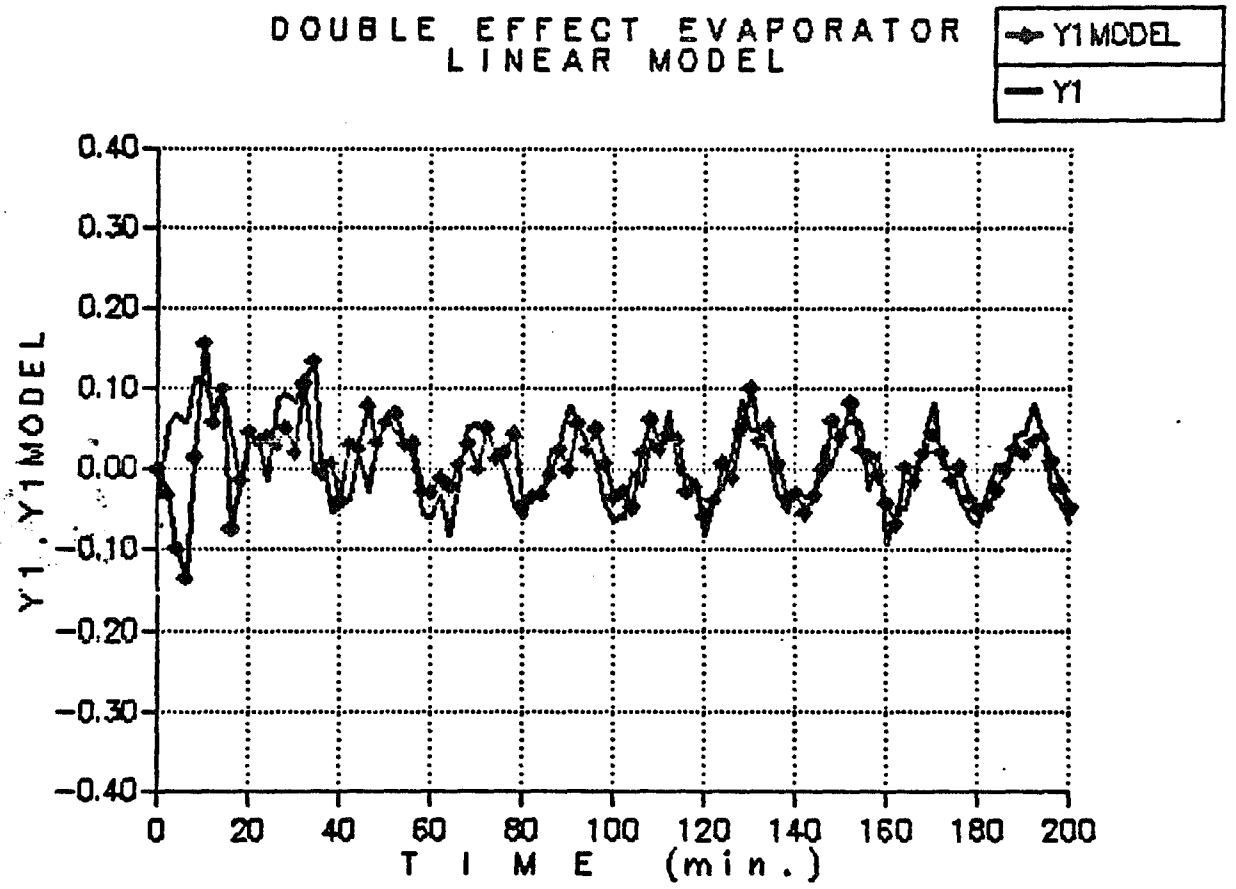

DOUBLE EFFECT EVAPORATOR
LINEAR MODEL
Y1 MODEL
Y1
0.40
0.30
0.20
0.10
0.00
-0.10
-0.20
-0.30
-0.40
Y1 , Y1MODEL
0
20
40
60
80
100
120
140
160
180
200
T I M E (min.)

Fig. 6.3

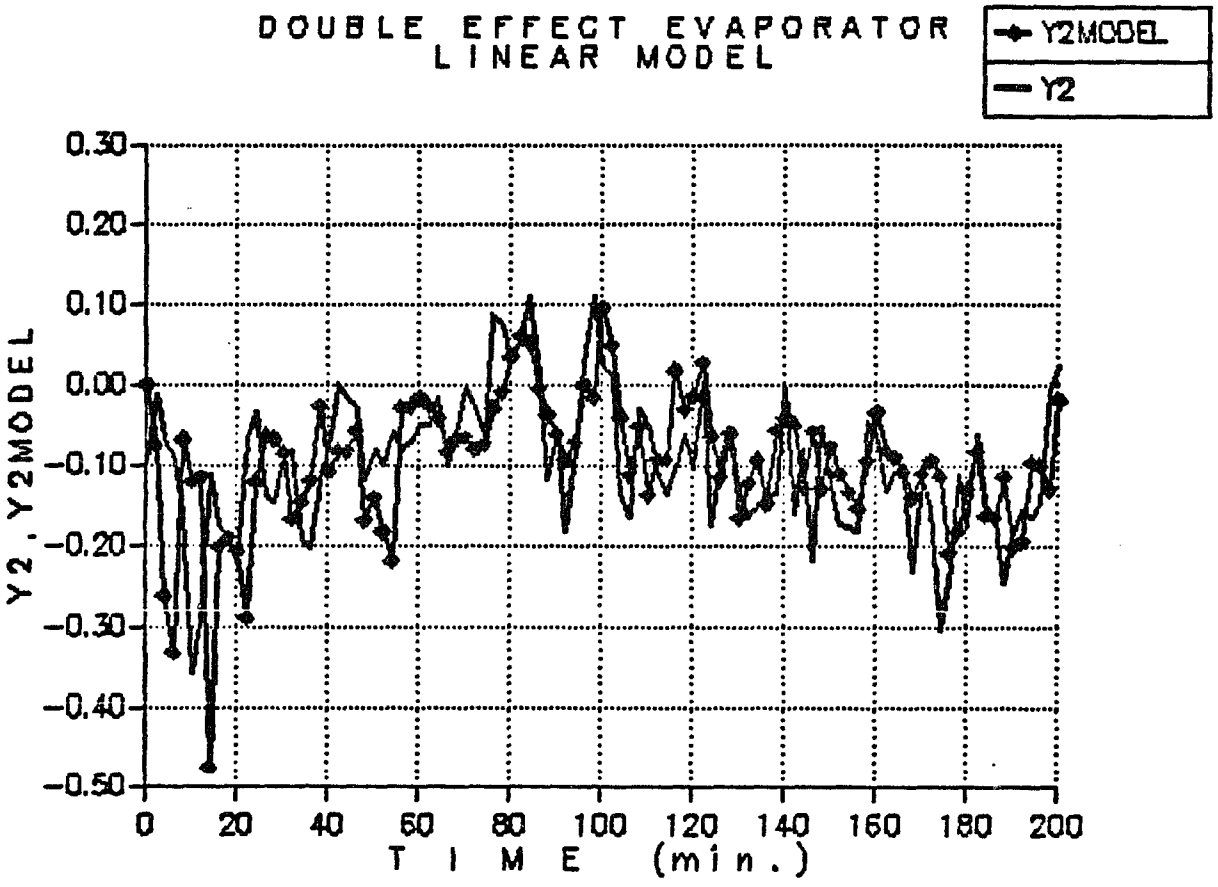

DOUBLE EFFECT EVAPORATOR
LINEAR MODEL
Y2MODEL
Y2
0.30
0.20
0.10
0.00
-0.10
-0.20
-0.30
-0.40
-0.50
Y2, Y2MODEL
0
20
40
60
80
100
120
140
160
180
200
T I M E (min.)

Fig. 6.4

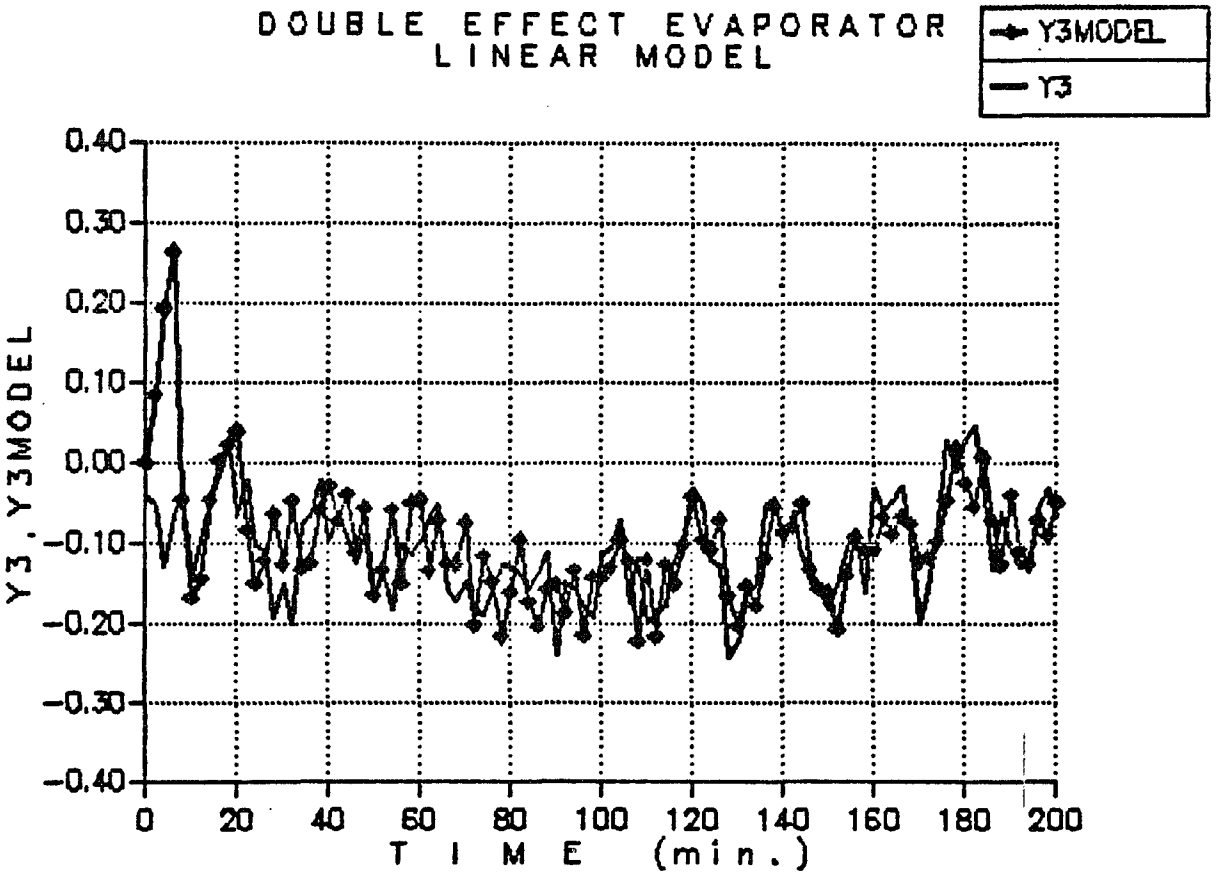

DOUBLE EFFECT EVAPORATOR
LINEAR MODEL
Y3MODEL
Y3
0.40
0.30
0.20
0.10
0.00
-0.10
-0.20
-0.30
-0.40
Y3, Y3MODEL
0
20
40
60
80
100
120
140
160
180
200
T I M E (min.)

Fig. 6.5

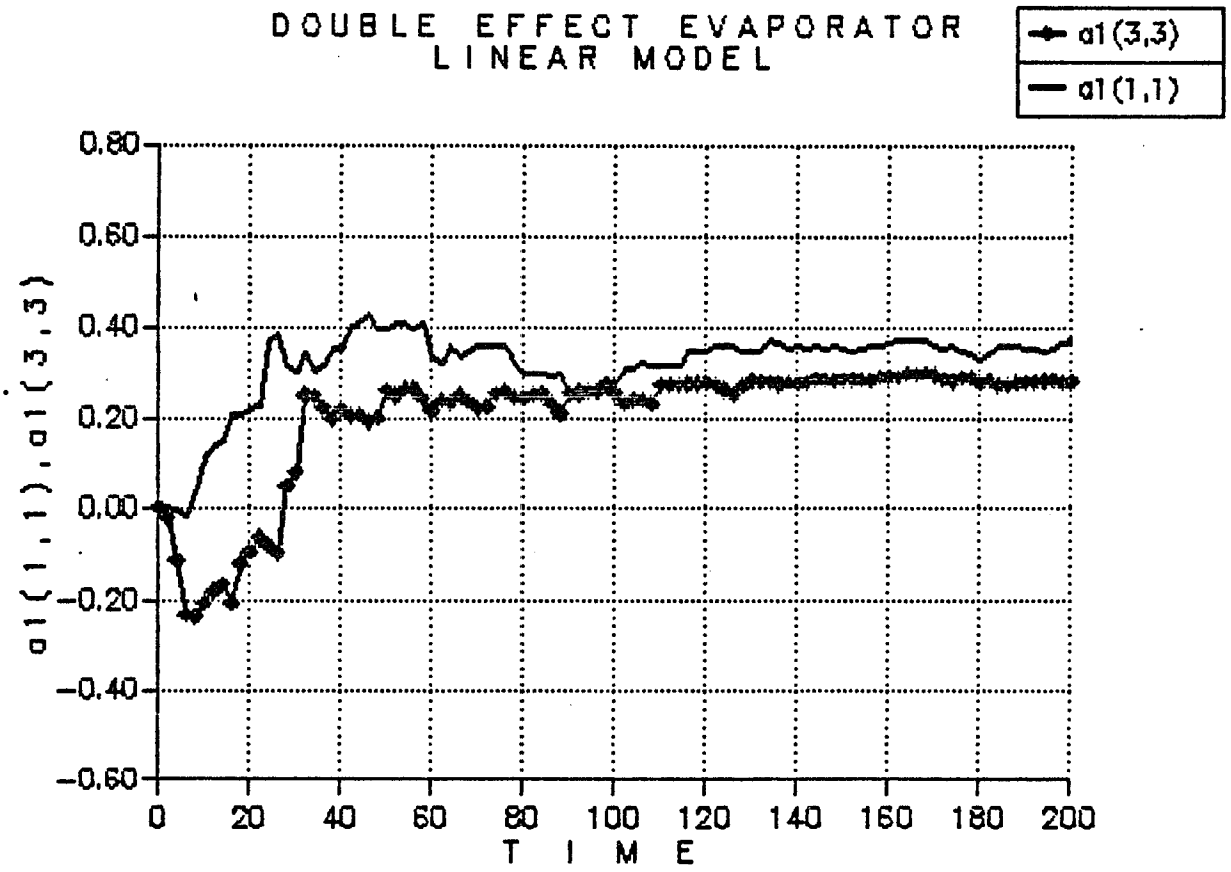

DOUBLE EFFECT EVAPORATOR
LINEAR MODEL
a1(3,3)
a1(1,1)
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
-0.60
a1(1,1),a1(3,3)
0
20
40
60
80
100
120
140
160
180
200
T I M E

Fig. 6.6

$$\dot{x}(5) = -0.0671x(5) + 0.2550F(t-8)$$

$$\dot{x}(6) = -0.0768x(6) + 0.3712F(t-3)$$

set 6.3

$y(1) = x(1)-x(2)+x(5)$ top composition

$y(2) = x(3)-x(4)+x(6)$ bottom composition

u1 = reflux flow rate

u2 = steam flow rate

F = feed flow rate

Ind = 0 if F = 0

Ind = 1 if F = 0

The purpose of Ind is to simulate a plant change when the feed flow rate is changed. It is assumed that the properties of the column most affected by feed flow changes are static gains and dead times. This is a reasonable assumption provided the dynamic lags (time constants) are less sensitive to flow changes in the plates and that no flooding occurs.

In terms of matrix transfer functions the system is represented by:

S1:

$$y(s) = \begin{bmatrix} \frac{12.8\ \exp(-s)}{16s+1} & \frac{-18.9\ \exp(-3s)}{21s+1} \\ \frac{6.6\ \exp(-7s)}{10.9s+1} & \frac{-19.4\ \exp(-3s)}{14.4s+1} \end{bmatrix} u(s) \qquad 6.4$$

S2:

$$y(s) = \begin{bmatrix} \frac{6.1\ \exp(-s)}{16s + 1} & \frac{-\ 8.4\ \exp(-2s)}{21s + 1} \\ \frac{3.5\ \exp(-4s)}{10.9s + 1} & \frac{-\ 8.8\ \exp(-s\ )}{14.4s + 1} \end{bmatrix} u(s)$$

$$\begin{bmatrix} \frac{3.8\ \exp(-8s)}{14.9s + 1} \\ \frac{4.9\ \exp(-4s)}{13.2s + 1} \end{bmatrix} F(s) \qquad 6.5$$

The time constants are in minutes.

Simulated plant : Set of equations 6.3. Runge Kutta integration with fixed step of 0.125 min.

Sampling period h: 1 min.

Input vector : white noise with zero mean and variances (0.025, 0.02)

Initial information matrix $P(0) = 10^7\ I$

Window parameter $\sigma o = 0.005$

Initial estimation $\theta(0) = 0$

Model order

Fig. 6.7 presents the average estimation error for different values of parameters N_{AM} and N_{BM}. The average error was defined as:

$$(\ e\ ) = \sum_{1}^{N_S} (e_K^2(1) + e_K^2(2)) / N_S$$

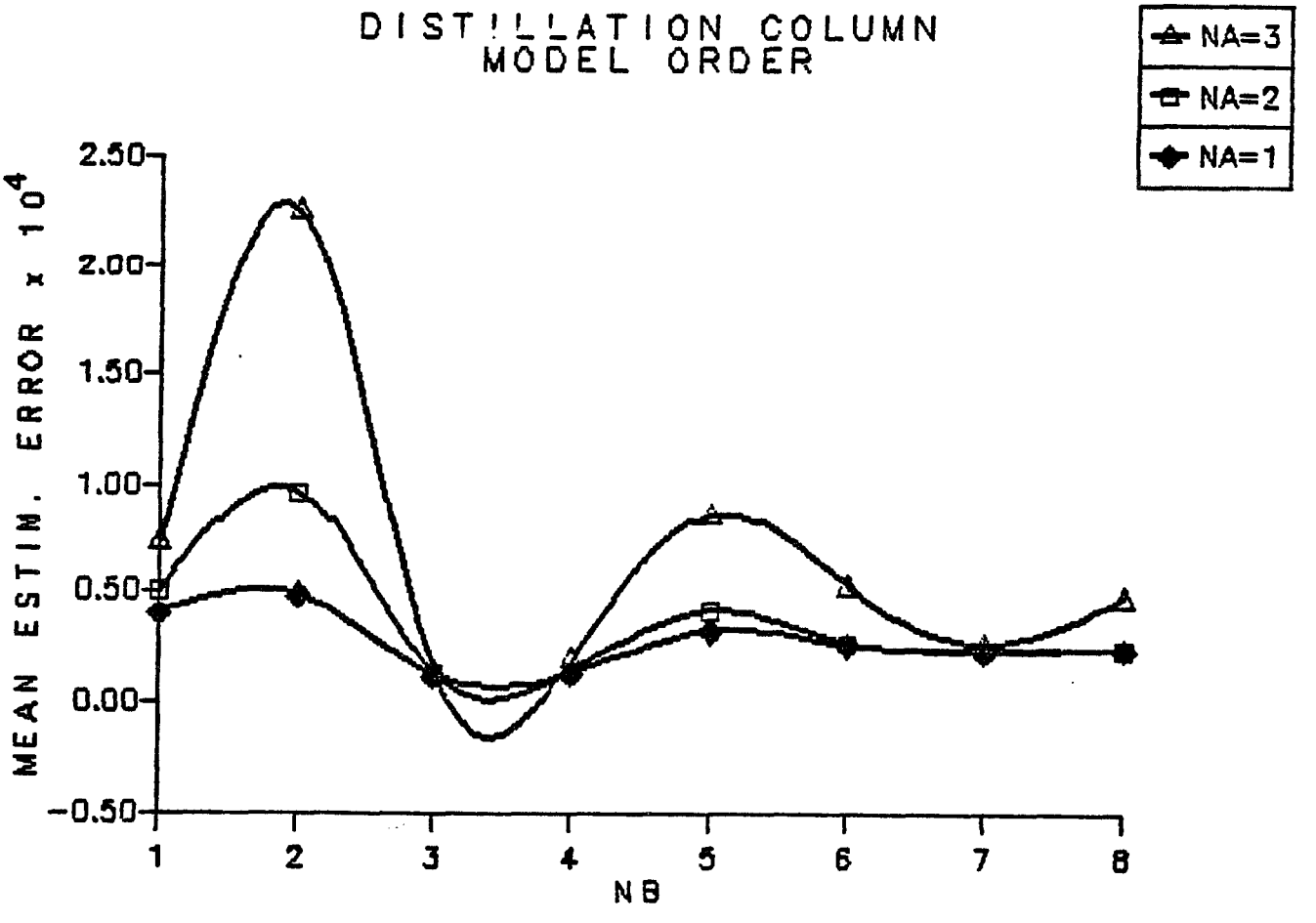

DISTILLATION COLUMN
MODEL ORDER
NA=3
NA=2
NA=1
MEAN ESTIM. ERROR x 10^4
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
1
2
3
4
5
6
7
8
NB

Fig. G.7

The smooth curves should be understood as plotter characteristics. Only the integer values make sense.

The models will be denoted Cij, C for column and i, j for the parameters N_{AM} and N_{BM} respectively.

The local minimum estimation errors are obtained for i = 1,2,3 and j = 3;i = 1,2 and j = 4,6;i = 1,2, 3 and j = 7. That is, there are at least 10 models with similar estimation error. Obviously, model C14 must be preferred to model C37 for example, that gives similar average error but needs 40 parameters instead of 20.

Open loop response

Figs.6.8 and 6.9 show the open loop response of the process together with the model output. The model is estimated on-line.

The vector y(k) represents the output at sampling time k.

The vector $y_m(k)$ represents the output of the discrete model G# at sampling time k. That is :

$$y_m = A^{-1} Bu = G\#u$$

where A and B are the estimated polynomial matrices, and u is the same input vector that drives the plant.

Models C13, C14, C16 and C27 were tested and no perceptible difference can be observed for the top composition y(1). For bottom composition y(2), models C13, C14 and C16 show small differences between y(2) and ym2.

DISTILLATION COLUMN
OPEN LOOP RESPONSE

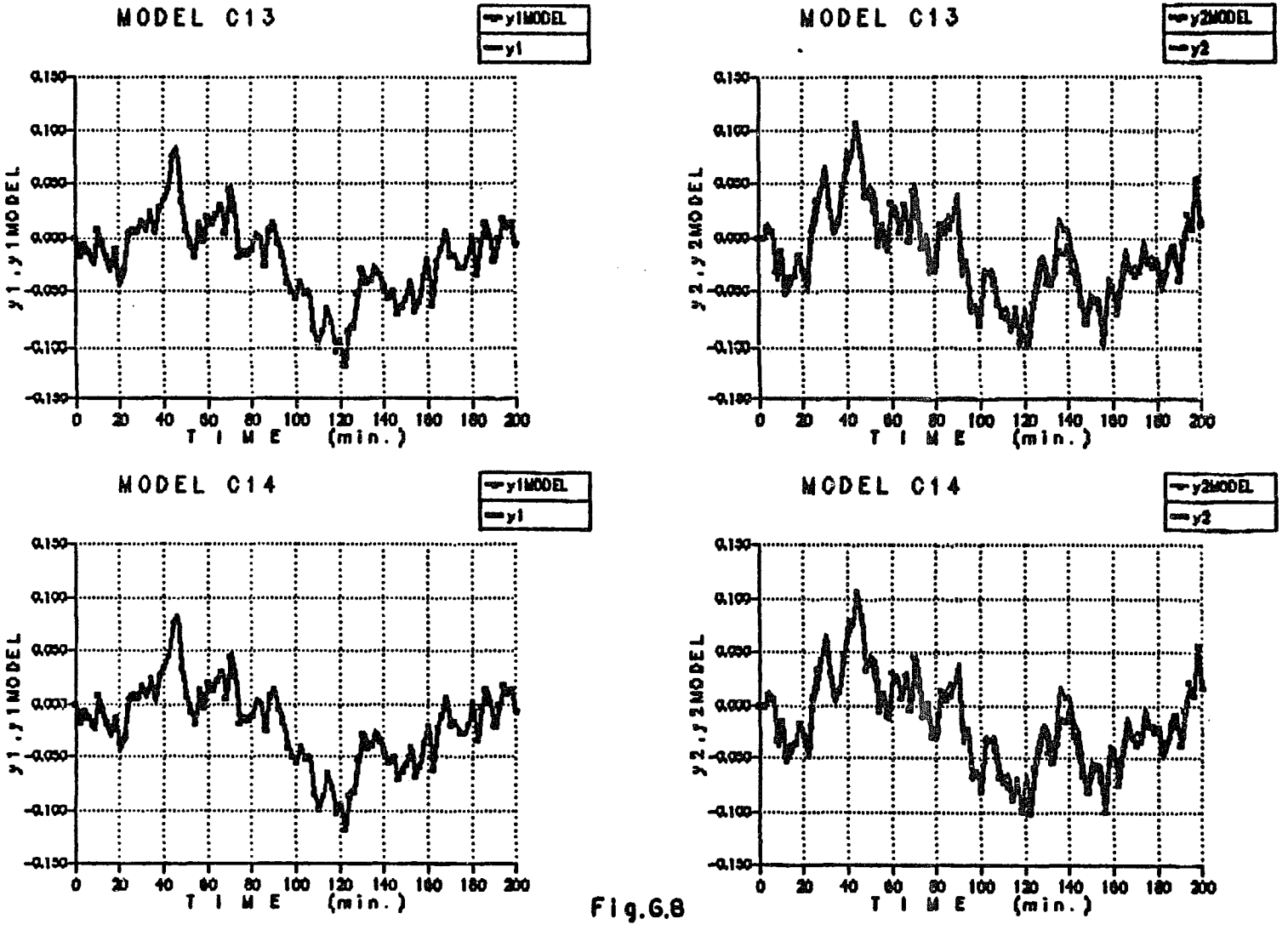


Fig.6.8

DISTILLATION COLUMN
OPEN LOOP RESPONSE

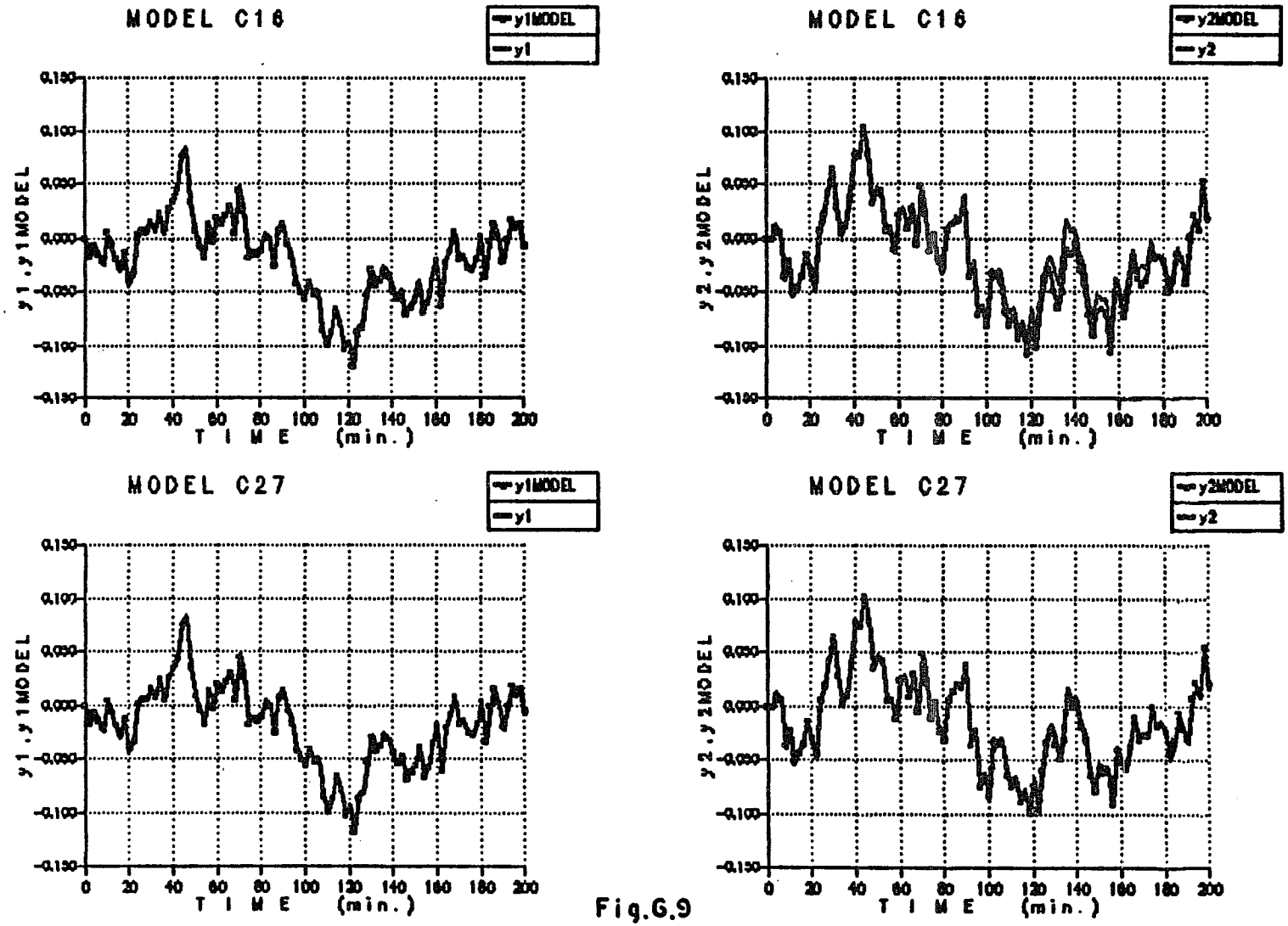


Fig.G.9

6.3.- CONTROL APPLICATIONS

A standard procedure was established for testing the control algorithm. We assumed the minimum knowledge about the process and started all the simulations in an open loop form with the same parameters used in the previous section.

After a number of identification samples that is chosen according to the model size, the system was switched to control mode and the input vector u was generated through the IMC structure. The changes in the set point and perturbations were started once the system was working in automatic.

Two set of additional parameters were necessary:

1) Filter parameters αi

These parameters were set initially at 0.5 and then were changed by trial and error. Normal operating ranges were found between 0.3 and 0.8.

2) Estimation error threshold

An average open loop estimation error was calculated. After switching to automatic, this value was used as an index in such a way to by-pass the calculation of the control structure if the model has not been updated. This procedure improves the computation load.

6.3.1.- Paper machine headbox

The system B1 described in 6.1, although very simple, has some undesirable characteristics that make it difficult to control by conventional methods.

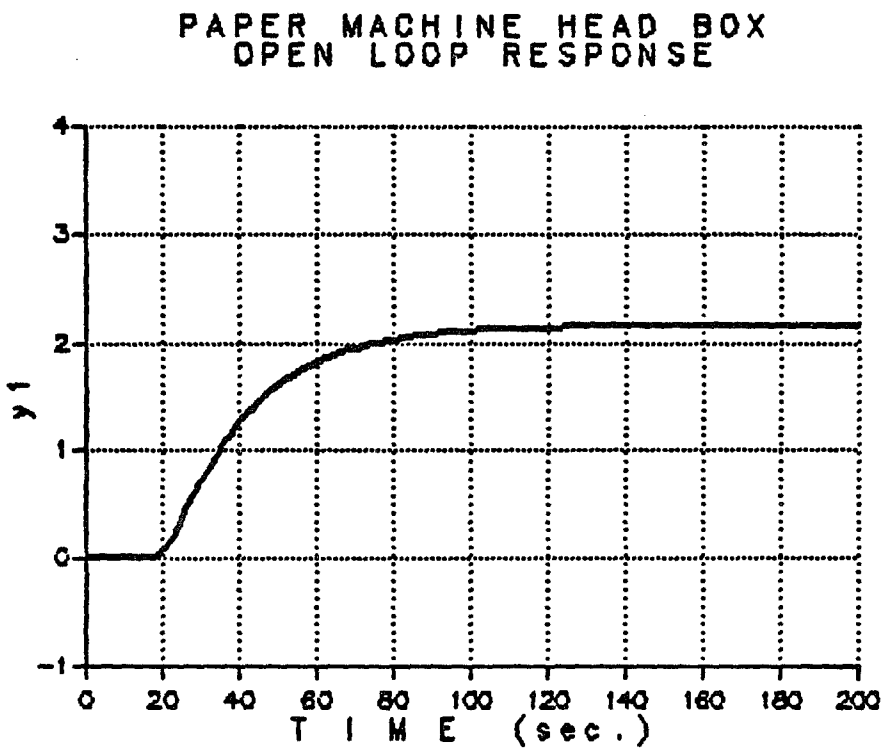
In Figs. 6.10a and 6.10b we can observe the open loop response when the process is forced to a new operation point by the application of a constant input vector u calculated as G(1) y* where y* is the new operating point. One of the output variable shows a very slow transient while the second output presents inverse response added to the slow transient.

In Figs.6.11 and 6.12 the performance of the adaptive multivariable internal model control is shown. During the first 20 samples the inputs to the system consisted of PRBS with amplitudes 0.01 and 0.02. The outputs were corrupted with white noise sequences with variance 0.010 and 0.020. Then the inputs were switched to those generated by the IMC algorithm and a sequence of set point changes was started:

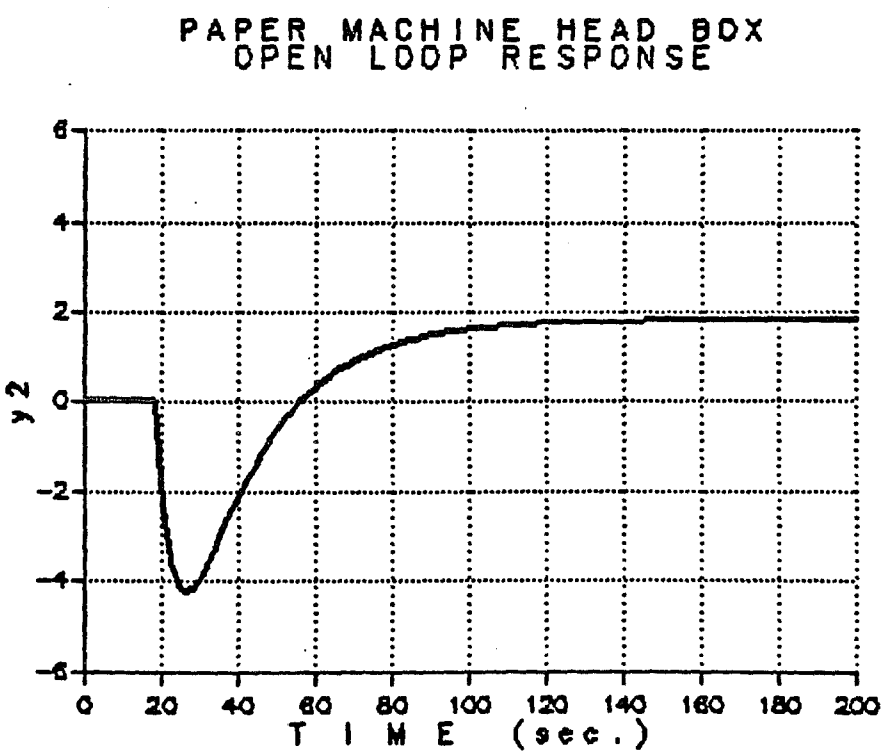
$$sp(1) = 2.15(r(19)-r(57)+r(82))$$
$$sp(2) = 1.80(r(25)-r(62)+r(82))$$

where r is the unit step function. The total decoupling and almost perfect set point tracking can be observed. Although the same behavior can be obtained with a fixed non adaptive controller designed following the internal model structure for a known plant, we must point out the complete self-tuning characteristic of our system.

Figs. 6.13 and 6.14 show the performance of the fixed non adaptive IMC system when the plant is changed from the


PAPER MACHINE HEAD BOX
OPEN LOOP RESPONSE
y 1
4
3
2
1
0
-1
0 20 40 60 80 100 120 140 160 180 200
T I M E (sec.)

a


PAPER MACHINE HEAD BOX
OPEN LOOP RESPONSE
y 2
6
4
2
0
-2
-4
-6
0 20 40 60 80 100 120 140 160 180 200
T I M E (sec.)

b

Fig. 6.10

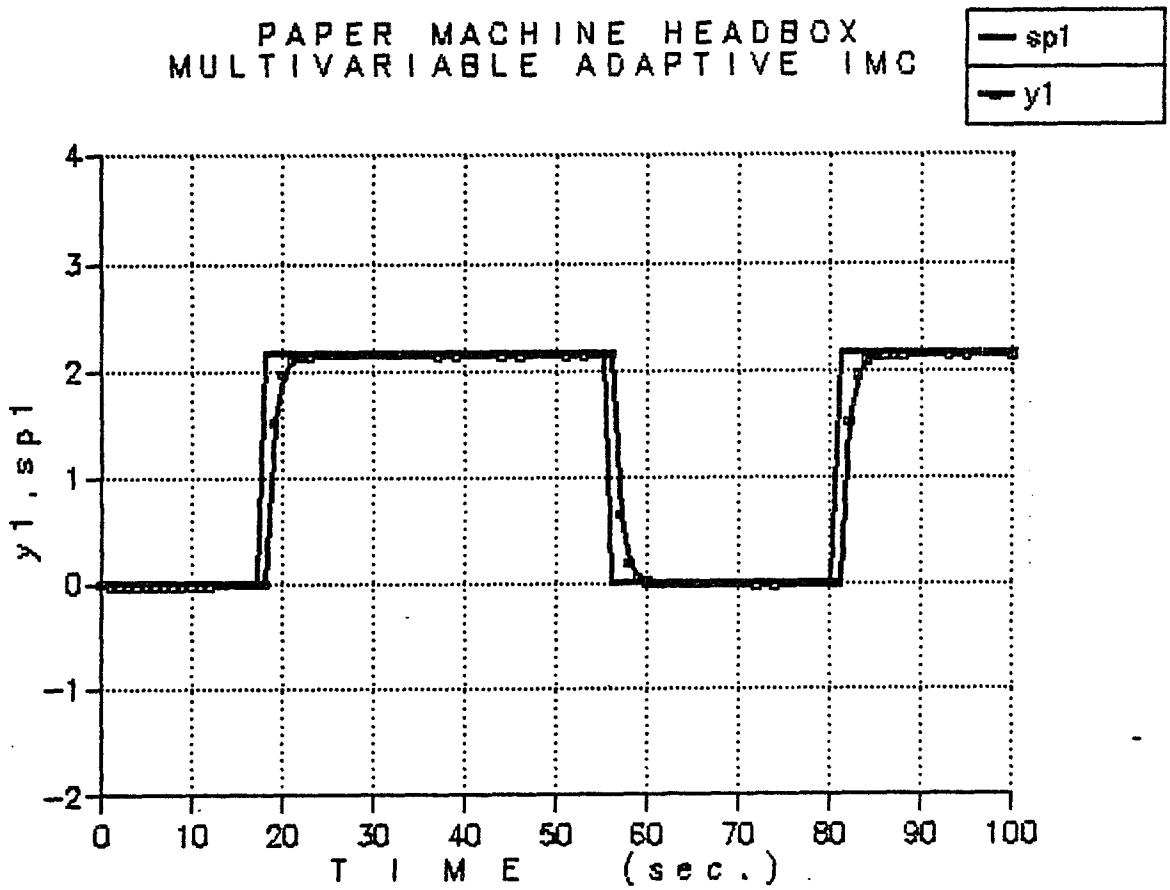

PAPER MACHINE HEADBOX
MULTIVARIABLE ADAPTIVE IMC
sp1
y1
y1 , sp1
4
3
2
1
0
−1
−2
0
10
20
30
40
50
60
70
80
90
100
T I M E (sec.)

Fig. 6.11

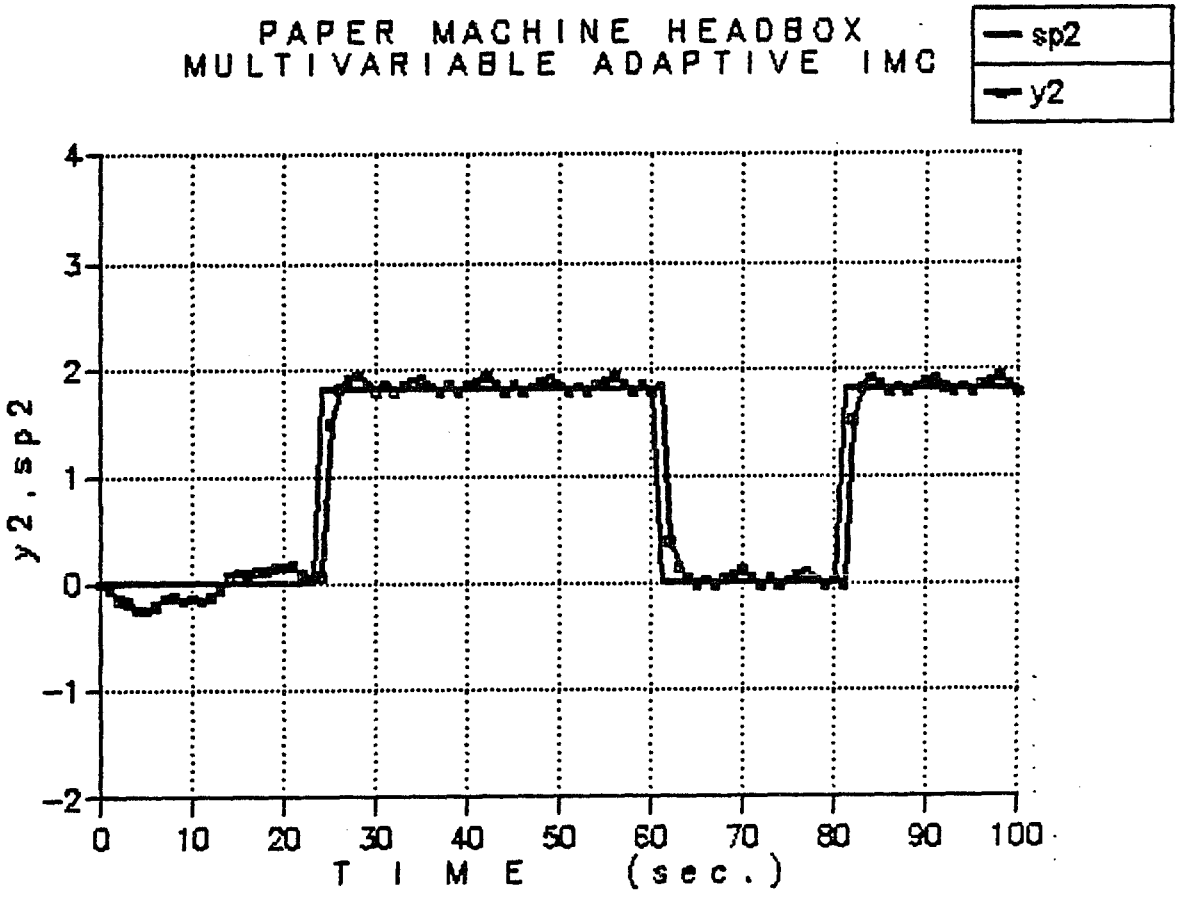

PAPER MACHINE HEADBOX
MULTIVARIABLE ADAPTIVE IMC
sp2
y2
y2 , sp2
4
3
2
1
0
−1
−2
0
10
20
30
40
50
60
70
80
90
100
T I M E (sec.)

Fig. 6.12

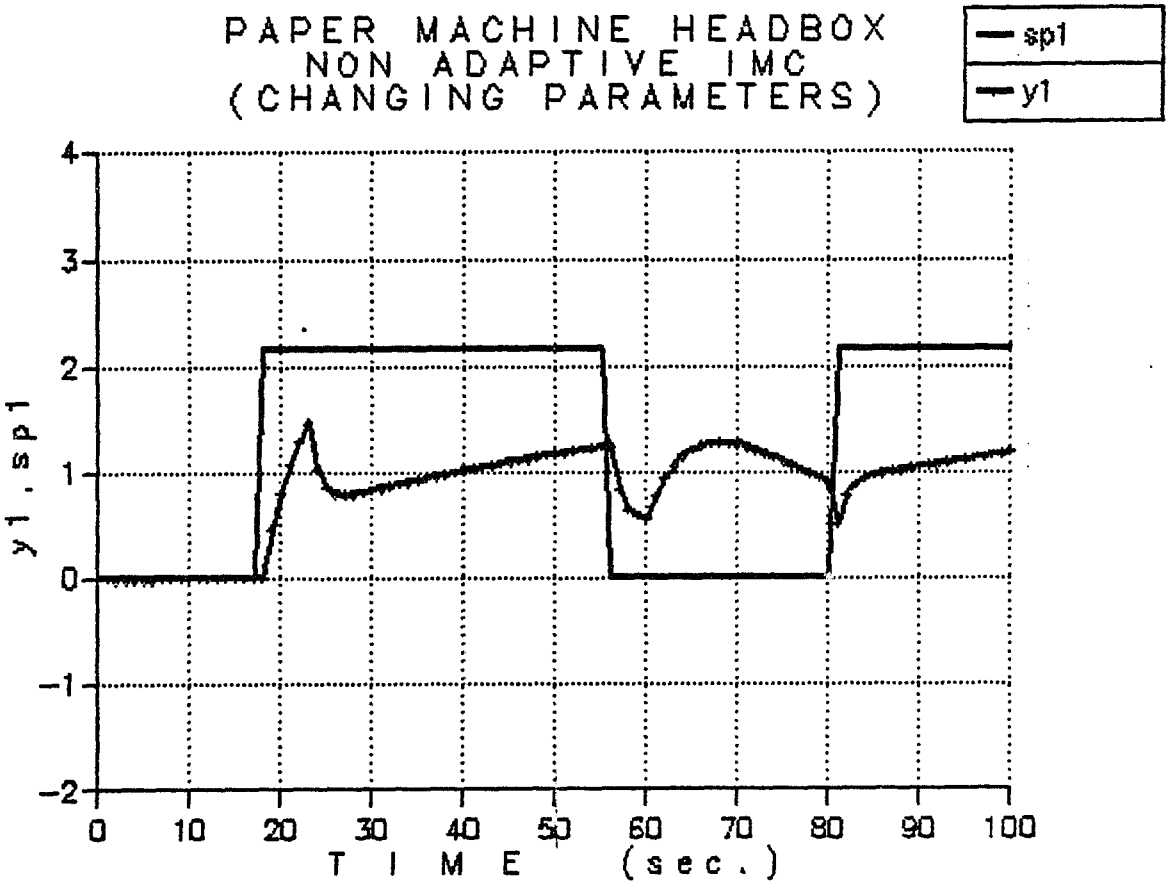

PAPER MACHINE HEADBOX
NON ADAPTIVE IMC
(CHANGING PARAMETERS)
sp1
y1
y1 , sp1
4
3
2
1
0
−1
−2
0
10
20
30
40
50
60
70
80
90
100
T I M E (sec.)

Fig. G.13

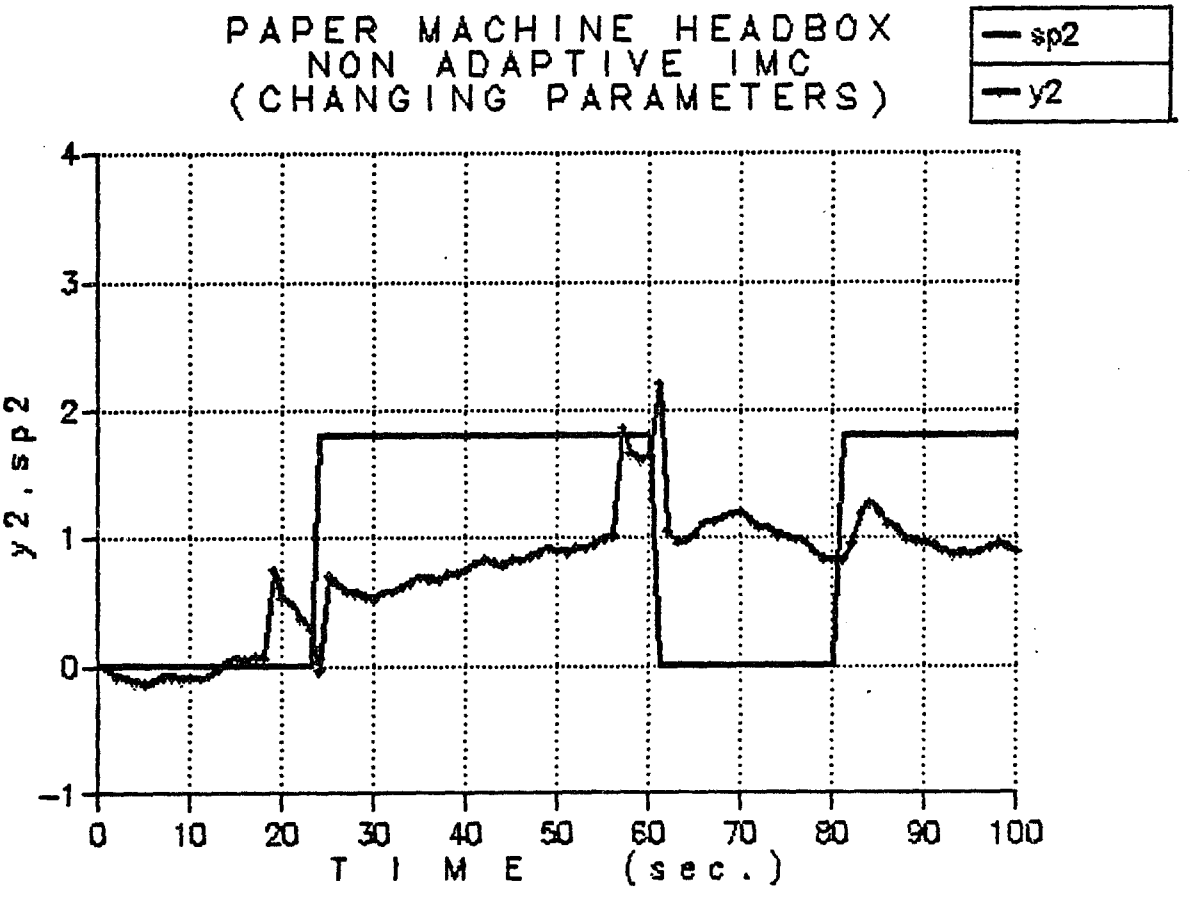
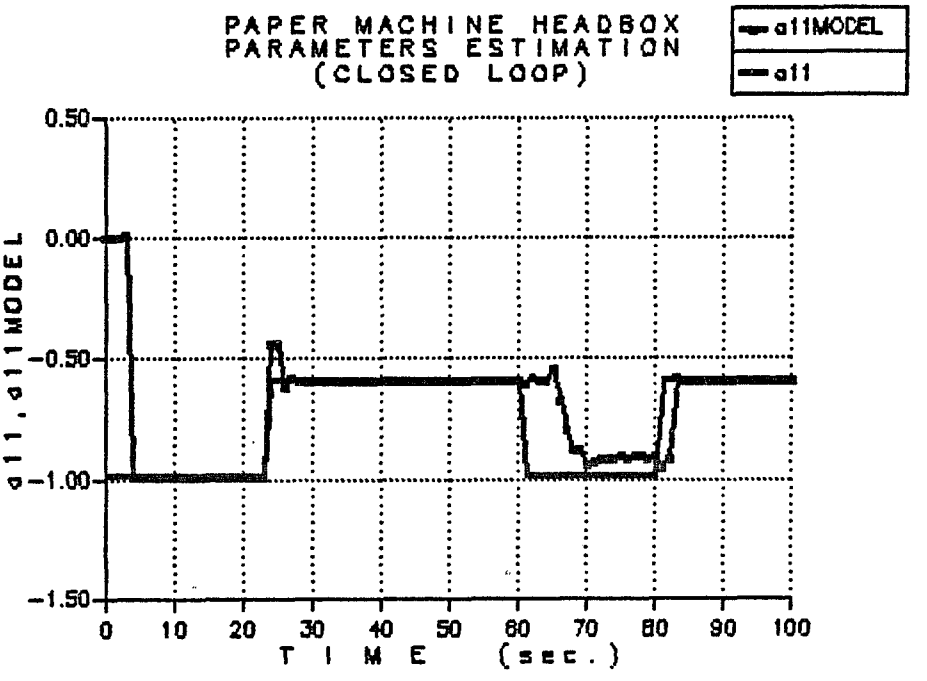

PAPER MACHINE HEADBOX
NON ADAPTIVE IMC
(CHANGING PARAMETERS)
sp2
y2
y2 , sp2
4
3
2
1
0
-1
0
10
20
30
40
50
60
70
80
90
100
T I M E (sec.)

Fig. 6.14

original system B1 to the new system B2. This change simulated a variation of plant parameters each time the total pressure (sp2) was moved to values different from zero. The change was achieved by decreasing the magnitude of a(1, 1) and b(2, 1). Although the change is moderate, the fixed IMC is completely deceived and the response is unsatisfactory.

We can point out that this is one weakness of the non adaptive internal model structure. According to the theory, condition (3.25) is accomplished and the global system is bounded input bounded output (BIBO) stable, but due to the parameters change, the process is non-linear and BIBO stability doesn't mean asymptotic stability any more. Therefore, the tracking properties are lost, and furthermore, this simple example shows that BIBO stability is a weak condition.

On the other hand, the adaptive multivariable IMC shows in Fig.6.15 that after some samples, the new system was identified with a performance equivalent to that obtained with open loop test (Fig.6.2). Therefore the control is recovered with satisfactory set point tracking as shown by Figs. 6.16a and 6.16b. Finally, Fig 6.17 shows the control requirement for the latter example. One of the manipulated variables, air flow u2, is for one sample far outside its normal range. We must recall that the system has been drastically forced to follow the set point, overcoming the slow dynamic and the inverse response characteristics. A better


PAPER MACHINE HEADBOX
PARAMETERS ESTIMATION
(CLOSED LOOP)
a11MODEL
a11
0.50
0.00
-0.50
-1.00
-1.50
a11, a11MODEL
0
10
20
30
40
50
60
70
80
90
100
TIME (sec.)

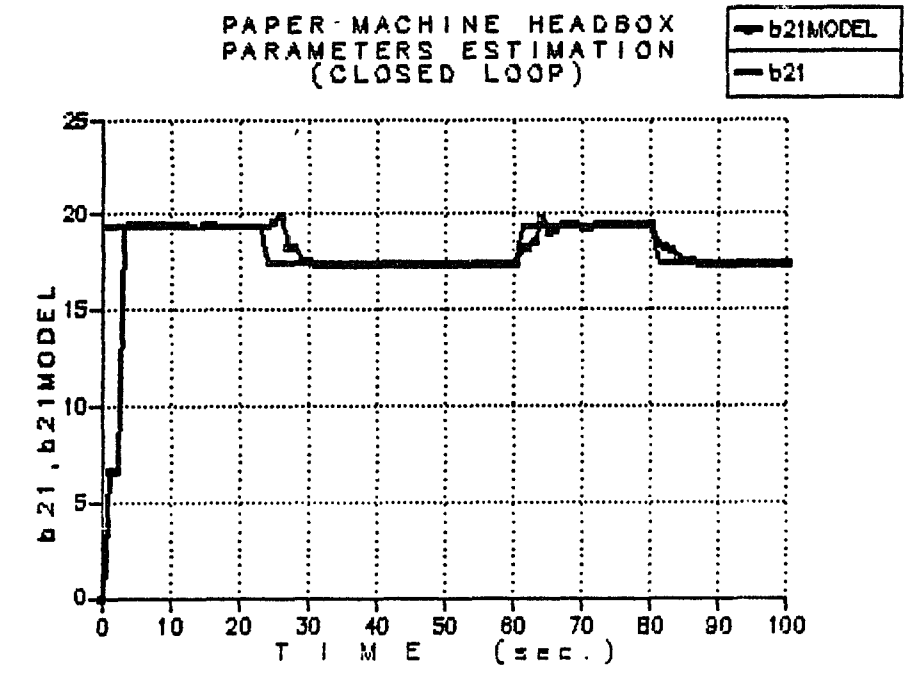
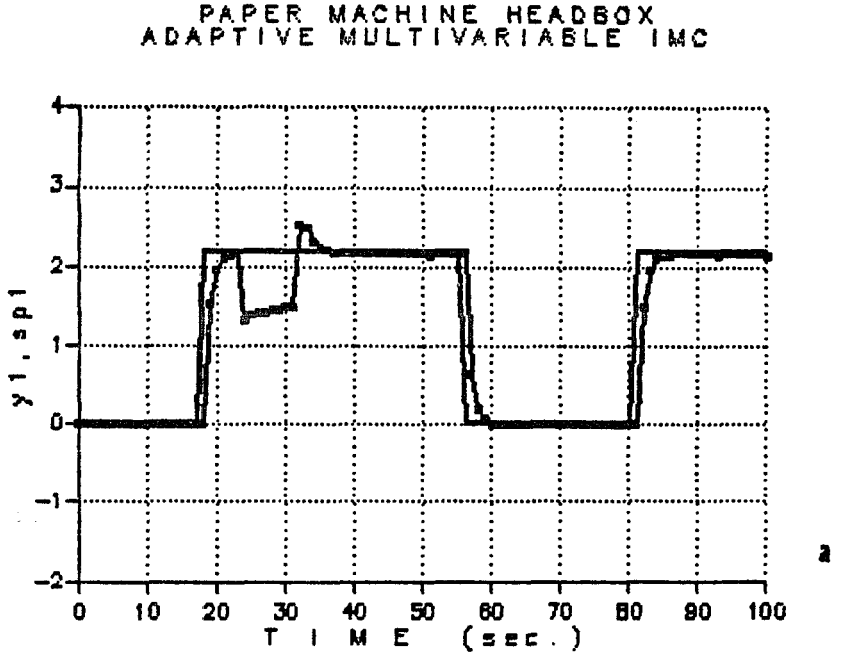

PAPER MACHINE HEADBOX
PARAMETERS ESTIMATION
(CLOSED LOOP)
b21MODEL
b21
25
20
15
10
5
0
b21, b21MODEL
0
10
20
30
40
50
60
70
80
90
100
TIME (sec.)

Fig. G.15


PAPER MACHINE HEADBOX
ADAPTIVE MULTIVARIABLE IMC
y1, sp1
4
3
2
1
0
−1
−2
0 10 20 30 40 50 60 70 80 90 100
T I M E (sec.)
a

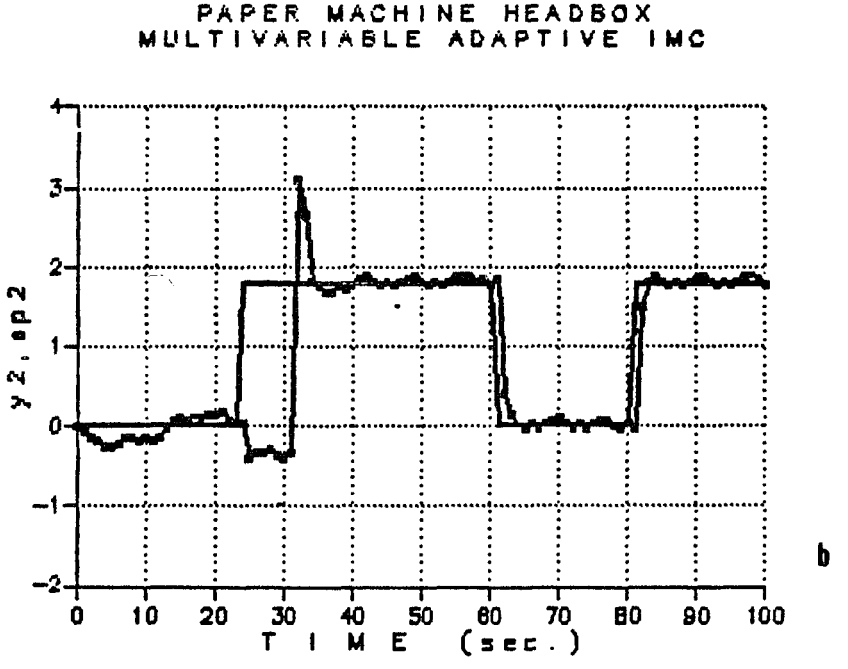

PAPER MACHINE HEADBOX
MULTIVARIABLE ADAPTIVE IMC
y2, sp2
4
3
2
1
0
−1
−2
0 10 20 30 40 50 60 70 80 90 100
T I M E (sec.)
b

Fig. G.IG

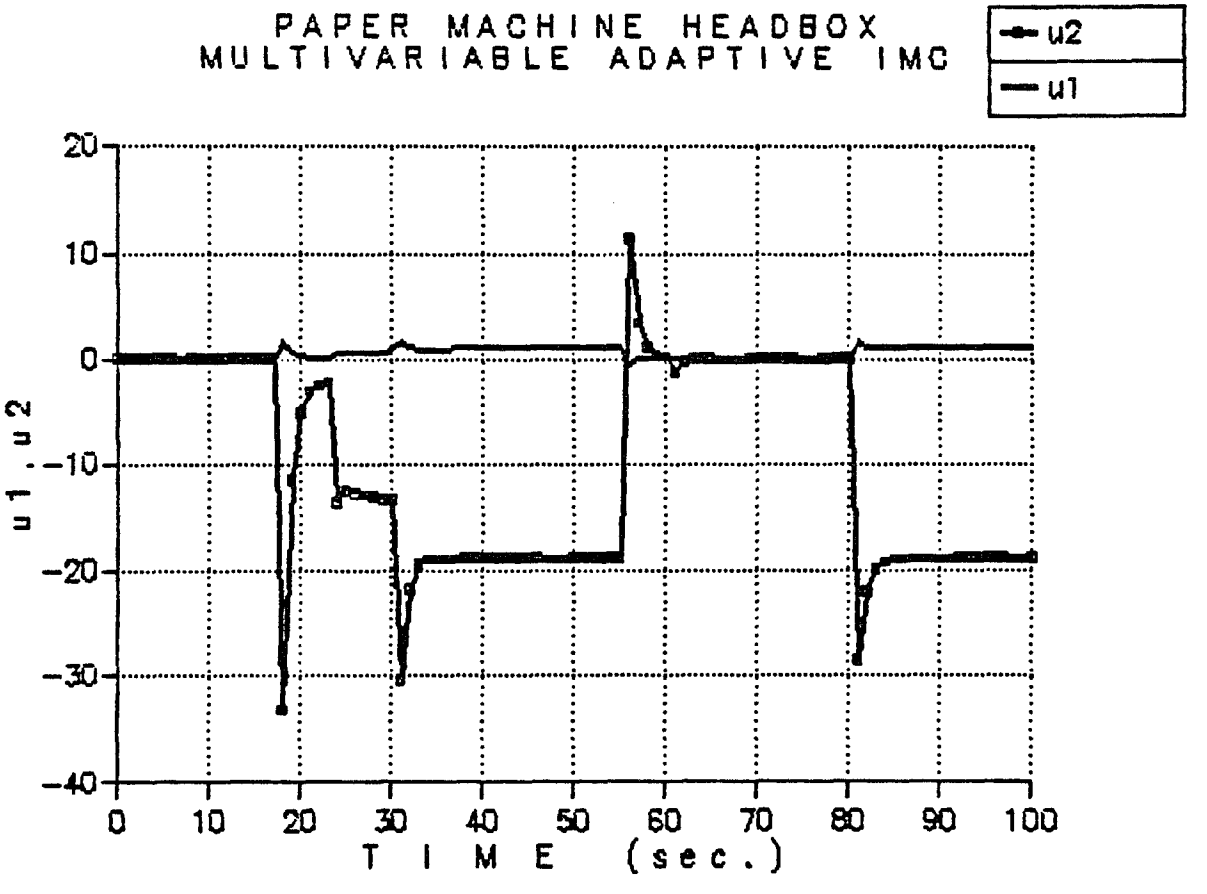

PAPER MACHINE HEADBOX
MULTIVARIABLE ADAPTIVE IMC
u2
u1
20
10
0
-10
-20
-30
-40
u1 , u2
0
10
20
30
40
50
60
70
80
90
100
T I M E (sec.)

Fig. 6.17

behavior for the manipulated variable can be obtained by increasing the filter parameters making the controlled system more sluggish.

6.3.2.- Distillation column

In the first place we tested the self-tuning IMC by introducing the following set point vector:

```
sp(1) = 0.8*r(30)
sp(2) = 1.6*(r(30)-r(155))
```

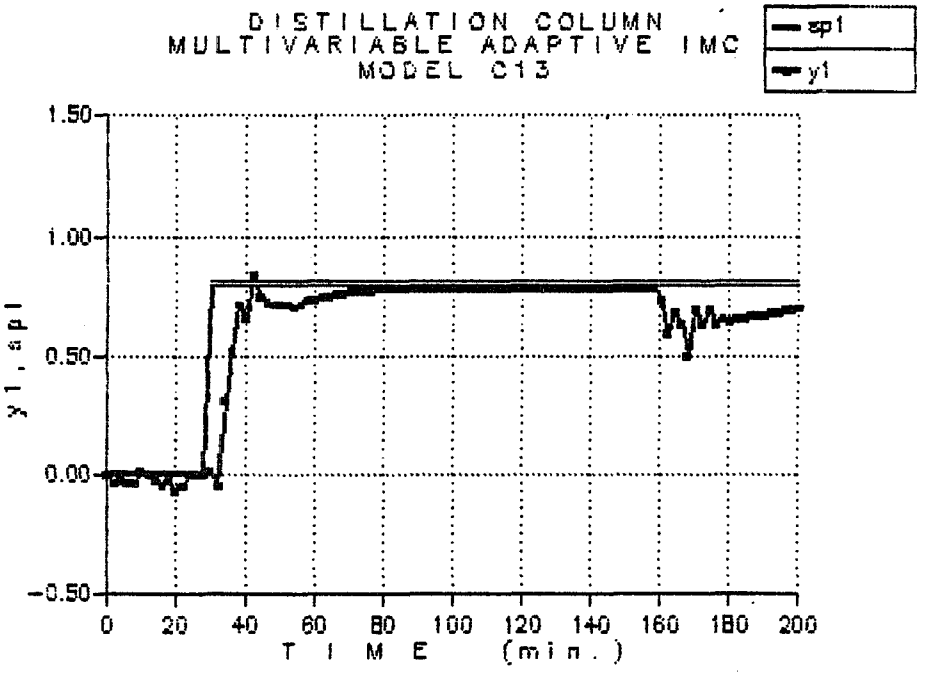
Then, the adaptive properties were tested by changing the plant characteristics from system S1 to system S2 after introducing a load disturbance at time = 80 min.and the following set point:

```
sp(1) = 0.8*(r(30)-r(57)+r(160))
sp(2) = 0.
F     = 0.34*r(80)
```

The control action was started after 30 samples and all the variables must be understood as deviations around a steady state.

Set point change

Figs.6.18 to 6.21 show the closed loop behavior for different models under adaptive internal model control. All


DISTILLATION COLUMN
MULTIVARIABLE ADAPTIVE IMC
MODEL C13
sp1
y1
1.50
1.00
0.50
0.00
-0.50
y1, sp1
0
20
40
60
80
100
120
140
160
180
200
T I M E (min.)

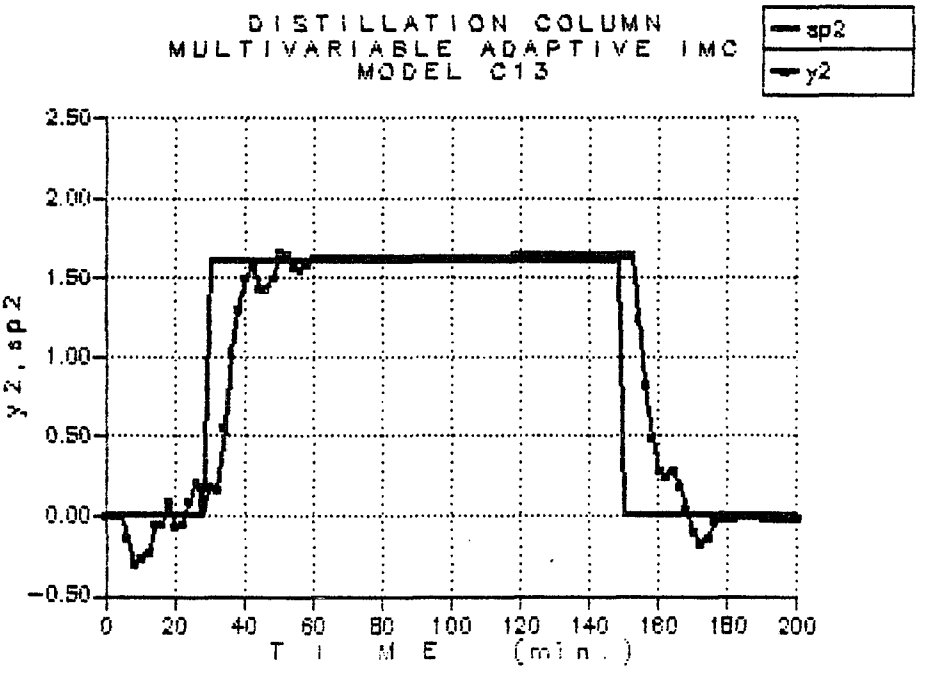

DISTILLATION COLUMN
MULTIVARIABLE ADAPTIVE IMC
MODEL C13
sp2
y2
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
y2, sp2
0
20
40
60
80
100
120
140
160
180
200
T I M E (min.)

Fig. 6.18

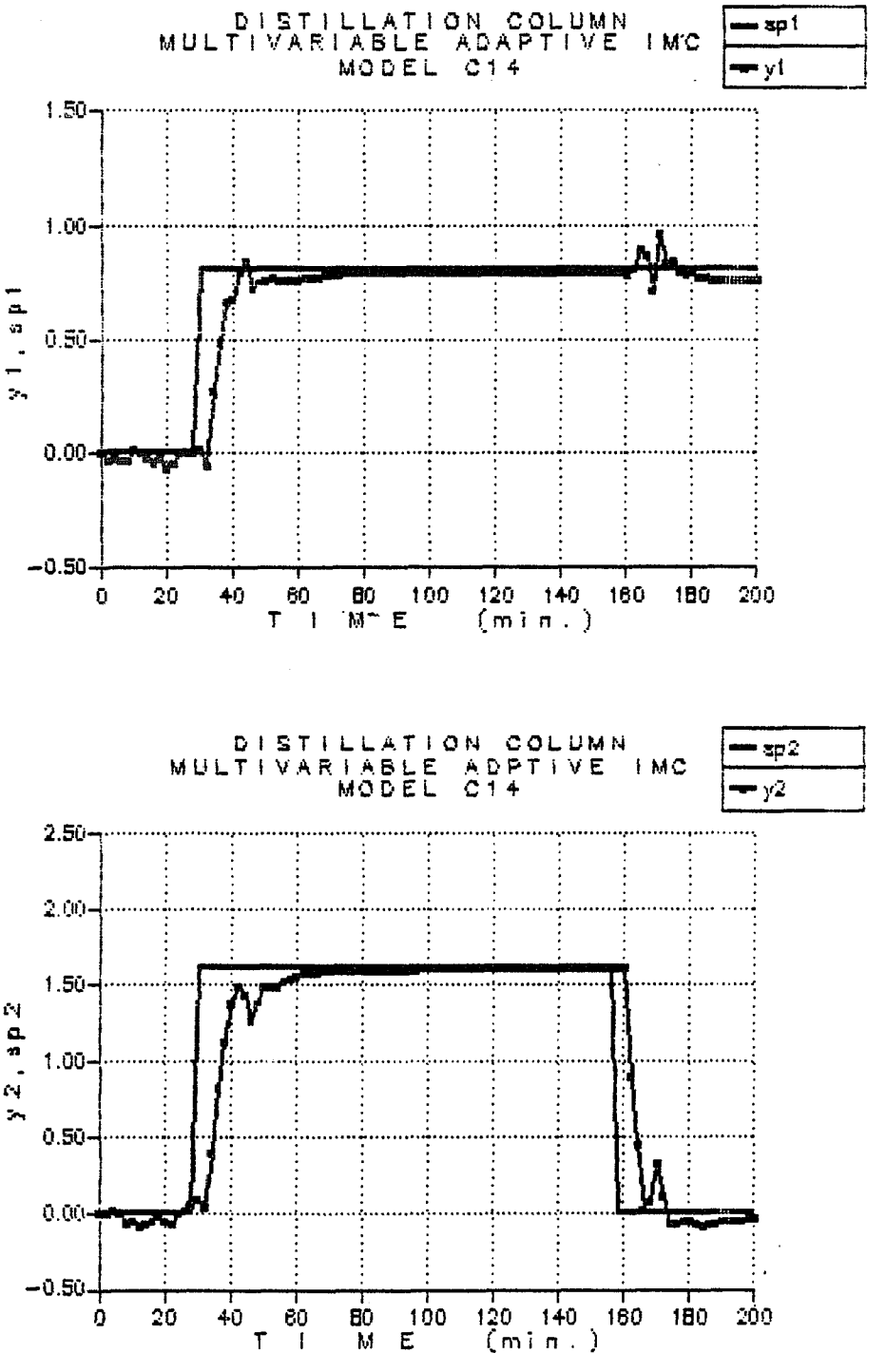
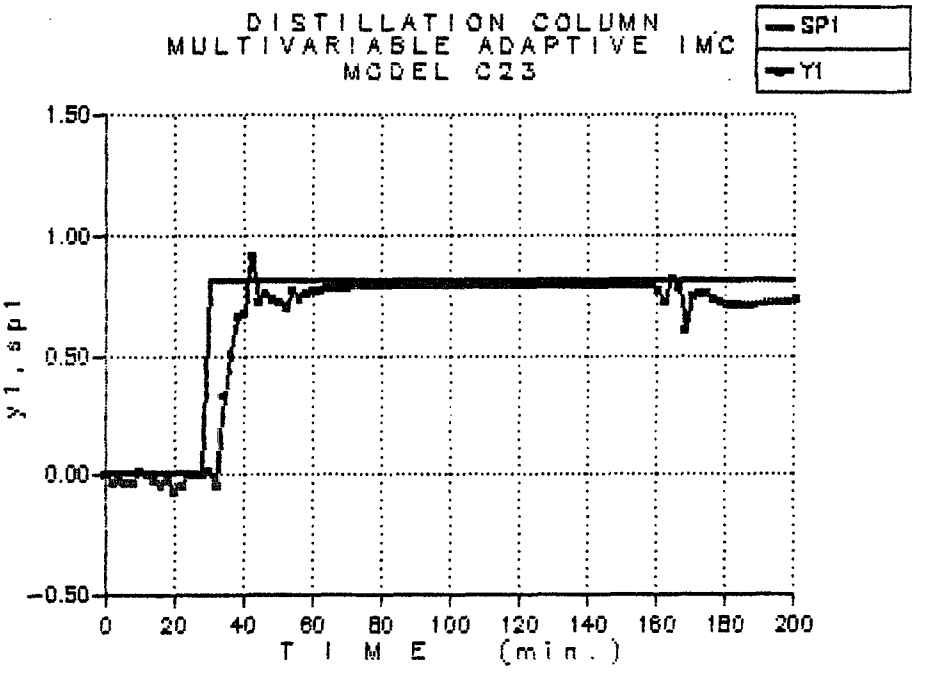

DISTILLATION COLUMN
MULTIVARIABLE ADAPTIVE IMC
MODEL C14
sp1
y1
1.50
1.00
0.50
0.00
-0.50
y1, sp1
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)
DISTILLATION COLUMN
MULTIVARIABLE ADPTIVE IMC
MODEL C14
sp2
y2
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
y2, sp2
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

Fig. G.19


DISTILLATION COLUMN
MULTIVARIABLE ADAPTIVE IMC
MODEL C23
SP1
Y1
1.50
1.00
0.50
0.00
-0.50
y1, sp1
0
20
40
60
80
100
120
140
160
180
200
T I M E (min.)

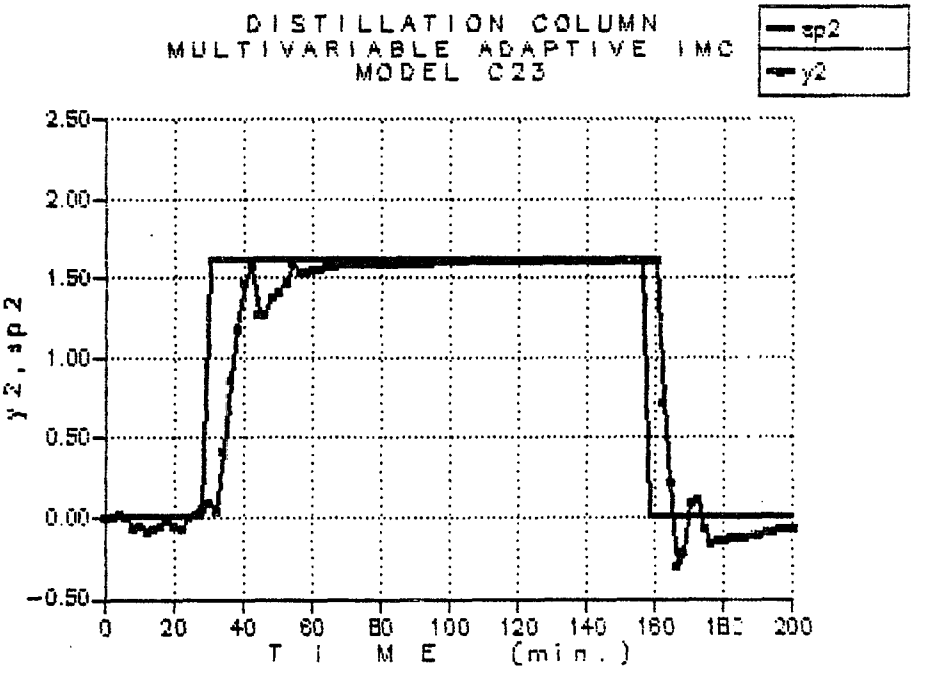
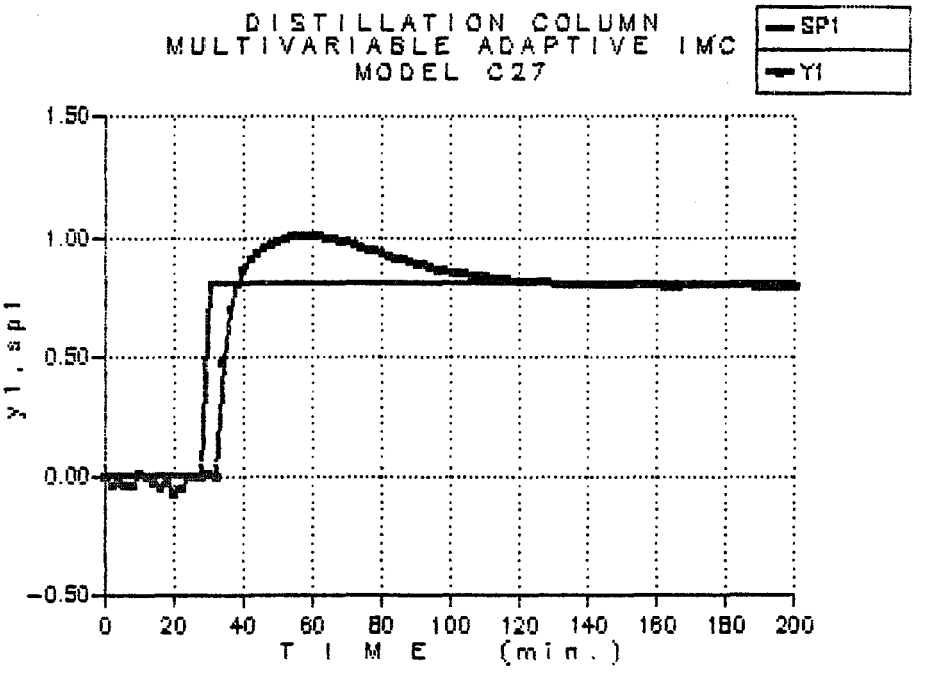

DISTILLATION COLUMN
MULTIVARIABLE ADAPTIVE IMC
MODEL C23
sp2
y2
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
y2, sp2
0
20
40
60
80
100
120
140
160
180
200
T I M E (min.)

Fig. 6.20


DISTILLATION COLUMN
MULTIVARIABLE ADAPTIVE IMC
MODEL C27
SP1
Y1
1.50
1.00
0.50
0.00
-0.50
y1, sp1
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

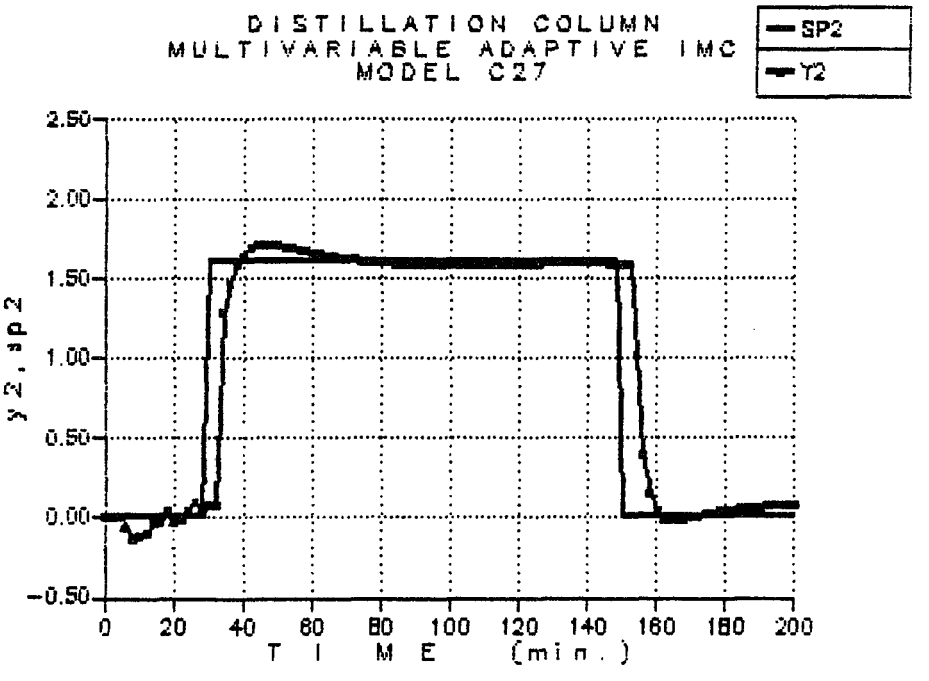

DISTILLATION COLUMN
MULTIVARIABLE ADAPTIVE IMC
MODEL C27
SP2
Y2
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
y2, sp2
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

Fig. G.21

the tested models lead to highly decoupled control. This decoupling and the smoothness of the response increase as the parameter N_{BM} increases.

The filter parameters were set to 0.60/0.70 for model C13 and C23 and 0.65/0.75 for models C14 and C27 (trial and error).

the estimation parameters were set at the same values used for the open loop test.

Load and system change

A step change in the feed stream was introduced. This sustained perturbation means also a change in the system dynamics that will be represented by S2 (6.5) instead of S1 (6.4).

The estimation problems that a deterministic load poses, can be explained through the following scheme:

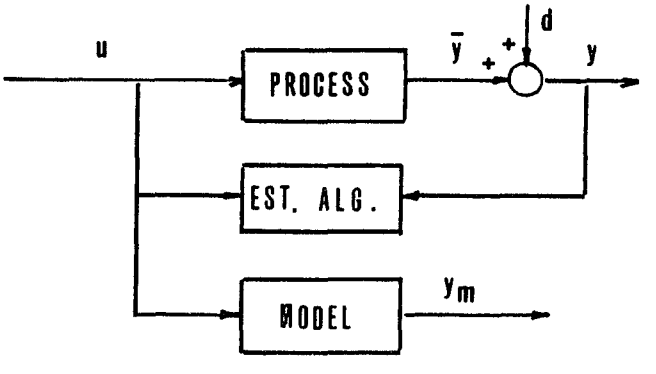


Fig. 6.22

Load Effects

The algorithm cannot establish the difference between a load disturbance or an eventual process parameters change (though in this test the change is more severe because the load also implies process gains and time delays modifications).
The load can be accounted for by the introduction of fictitious parameters to be estimated:

if $y = Gu + d = \bar{y} + d$

where d represents the load, then:

$\hat{y} = A^{-1}Bu + \hat{d}$ is the estimated output vector while $\hat{d}$ is the estimated load (do not confound $\hat{y}$ with y_m the model output).
$\hat{y}$ can be expressed as:

$$\hat{y} = \theta \phi \quad \text{with}$$

$$\theta = (A_1, \ldots, A_{N_{AM}}, B_1, \ldots, B_{N_{BM}}, \delta)^T \quad \text{and}$$

$$\phi^T = (y^T(t-1), \ldots, u^T(t-1), \ldots u^T(t-N_{BM}), 1)$$

Example
A simple single input single output example will help to understand the preceding statement:

$$\text{Let} \quad G(z^{-1}) = \frac{z^{-1} - 0.5z^{-2}}{1 - 1.4z^{-1} + 0.48z^{-2}}$$

and $y(k) = 1.4y(k-1)-0.48y(k-2)+u(k-1)-0.5u(k-2)+e+d$

where e is a white noise sequence and d is a deterministic load.

Fig. 6.23a shows the estimated parameters when the standard recursive least square algorithm (RLS) is used and no load affects the process.

At this stage, $\bar{y} = Gu$ and $y_m = G\#u$ differ only in the white noise since the estimated parameters are very close to the true parameters.

Fig 6.23b shows the same system with a load equal to 17.5% of the input amplitude. After the load is introduced, the estimated parameters are biased, y_m is no longer equal to $\bar{y}$ and a large modeling error is produced.

Fig. 6.23c shows the performance of the modified RLS with a fictitious parameter added to the estimation and a component of value one added to the regressor vector. This extra parameter lumps all the effects of the load in such a way that the other parameters are unchanged. So $y_m \tilde{=} \bar{y}$ and $y - y_m = \hat{d}$, estimated load. This estimated load is important because it corresponds to the feedback signal to the IMC controller. Of course, we cannot differentiate between load and modeling error, therefore if the model is a bad representation of the system, the feedback signal is contaminated. Our conjecture is that adapting the model to follow the plant changes, makes IMC more robust.

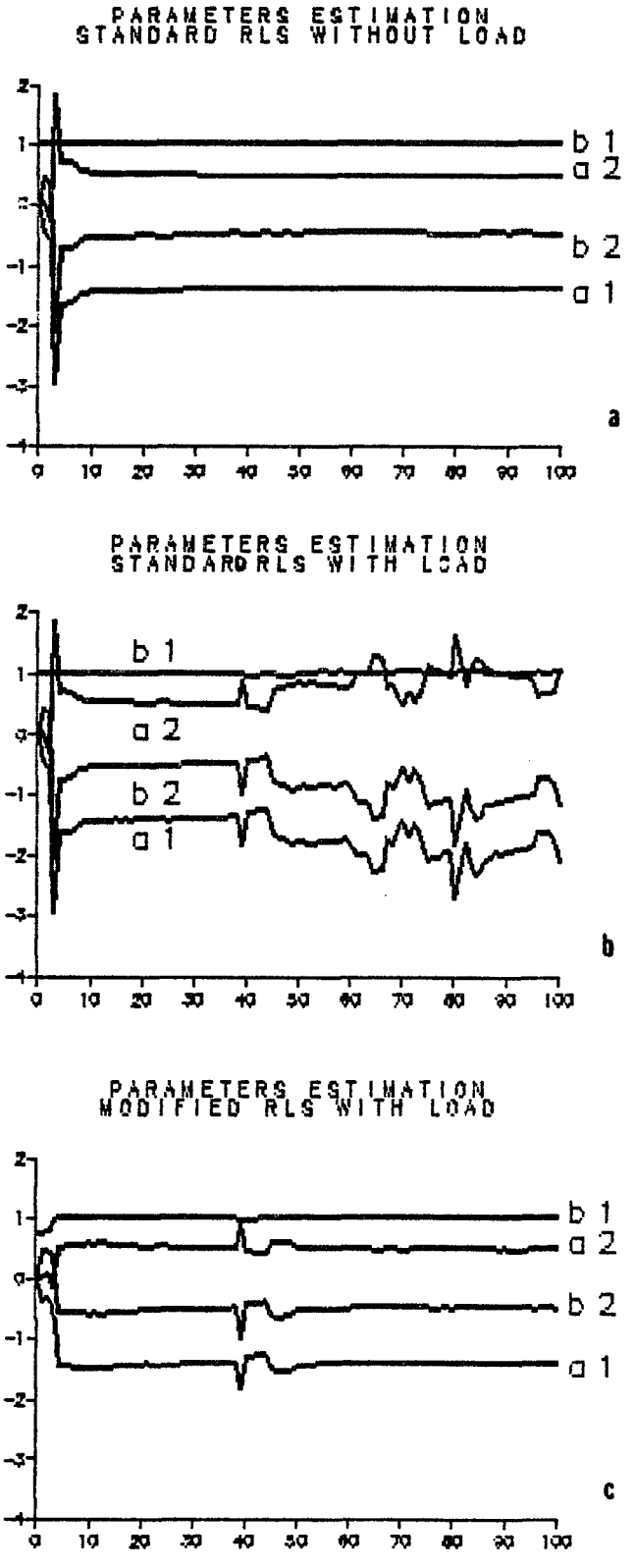

PARAMETERS ESTIMATION
STANDARD RLS WITHOUT LOAD
b 1
a 2
b 2
a 1
a
PARAMETERS ESTIMATION
STANDARD RLS WITH LOAD
b 1
a 2
b 2
a 1
b
PARAMETERS ESTIMATION
MODIFIED RLS WITH LOAD
b 1
a 2
b 2
a 1
c

Fig. 6.23

The changing parameters are detected with the use of a variable forgetting factor and making the algorithm more sensitive by decreasing the observation window (decreasing uo).

Fig. 6.24 shows the actual load $d = G_d F$ for a step change of 0.34 lbs/min in the feed stream. It also shows the difference $y-y_m$ ($\bar{d}$ following the IMC nomenclature). It is evident that the load identification is satisfactory.

In the next test , a different sequence of changes in set point was used, with sp2 = 0. This is a known difficult test (see McDermott and Mellichamp (62)) because of the lack of persistent information for one of the inputs. At time 80 minutes, the step load was introduced and the process was changed to S2. Fig. 6.25 shows how the proposed control was able to overcome the load effects and the system modifications. By the way, the slow rejection of the load is a problem proper of the IMC structure whenever the system dynamics is slow (the transient corresponds to the open loop response). This problem was pointed out by Svoronos (83) while working with single input single output systems.

Effect of persistent signals

Figs. 6.26,6.27,6.28, and 6.29 show the performance of the proposed adaptive internal model control when the process is forced to a new operating point by changing both set points and at the same time a sustained load is applied at

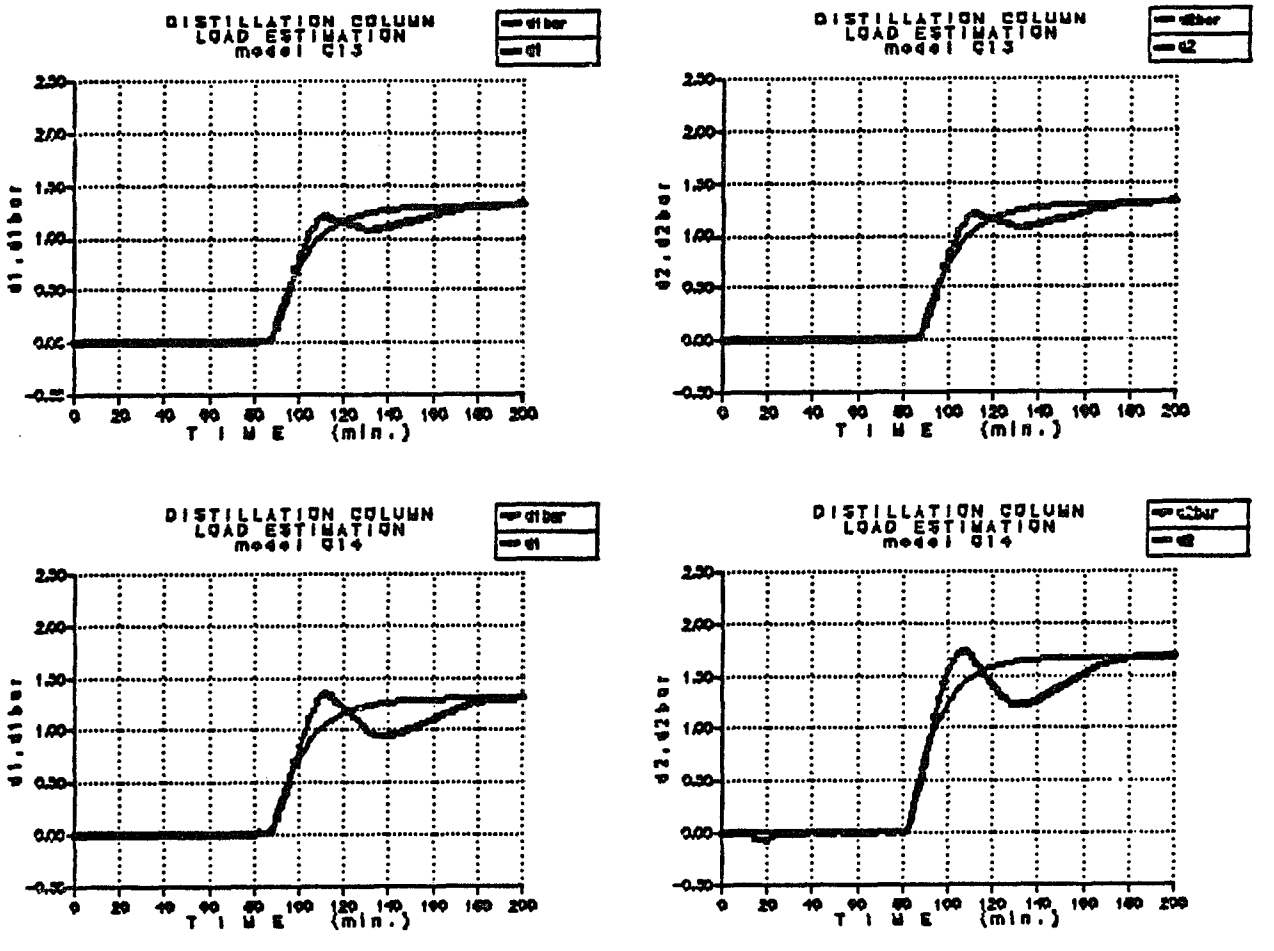
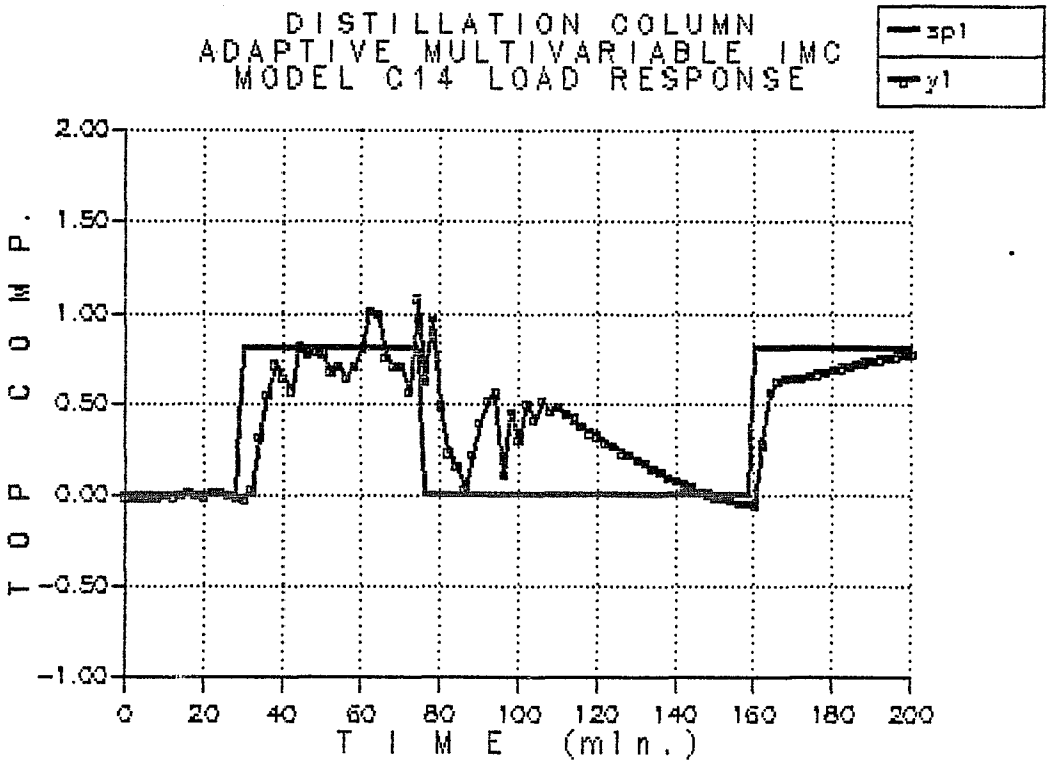

DISTILLATION COLUMN
LOAD ESTIMATION
model C13
d1bar
d1
d1, d1bar
TIME (min.)
DISTILLATION COLUMN
LOAD ESTIMATION
model C13
d2bar
d2
d2, d2bar
TIME (min.)
DISTILLATION COLUMN
LOAD ESTIMATION
model C14
d1bar
d1
d1, d1bar
TIME (min.)
DISTILLATION COLUMN
LOAD ESTIMATION
model C14
d2bar
d2
d2, d2bar
TIME (min.)

Fig. 6.24


DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C14 LOAD RESPONSE
sp1
y1
TOP COMP.
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
0
20
40
60
80
100
120
140
160
180
200
TIME (mln.)

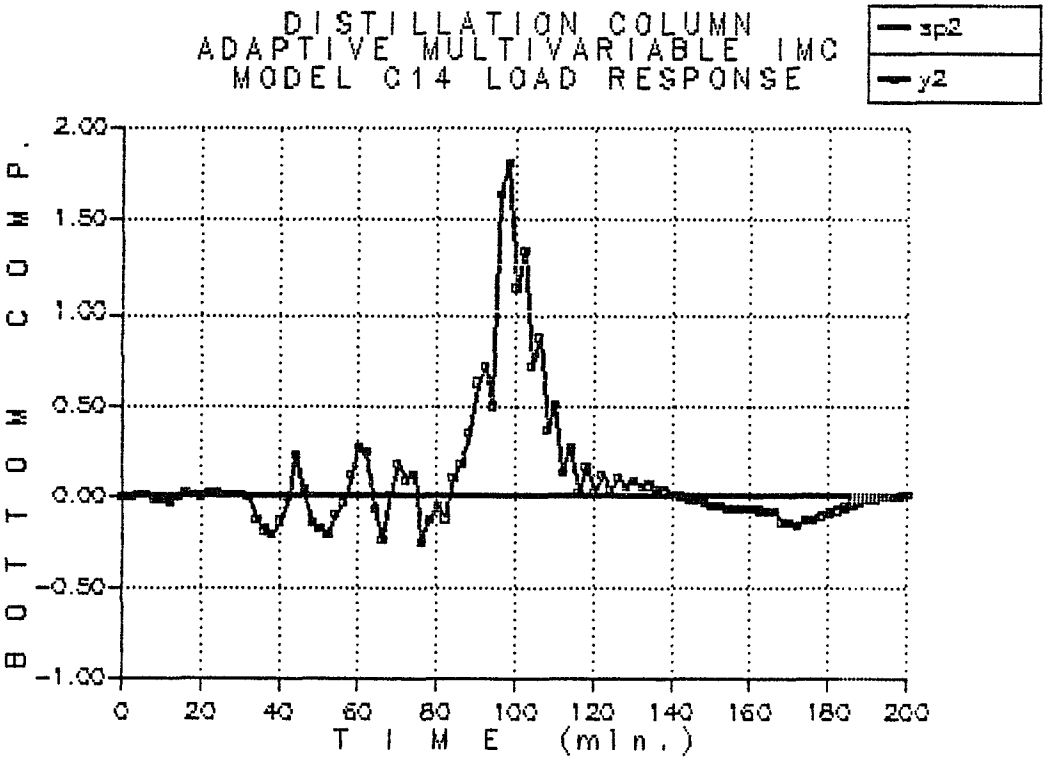
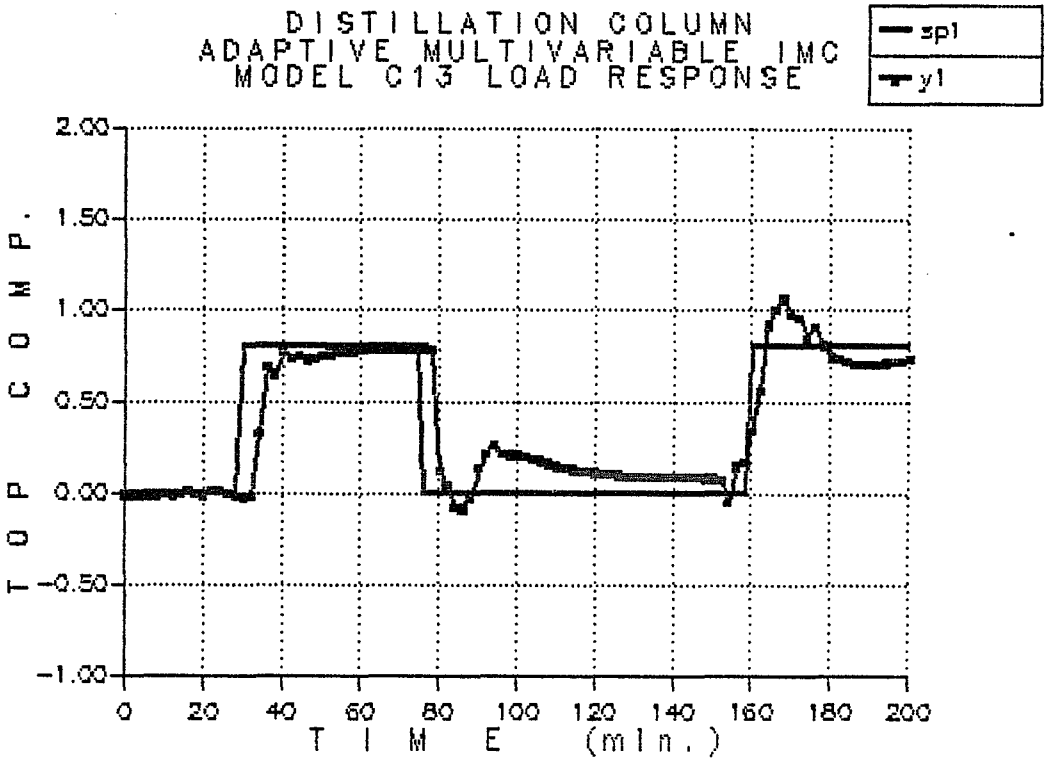

DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C14 LOAD RESPONSE
sp2
y2
BOTTOM COMP.
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
0
20
40
60
80
100
120
140
160
180
200
TIME (mln.)

Fig. 6.25


DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C13 LOAD RESPONSE
sp1
y1
TOP COMP.
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

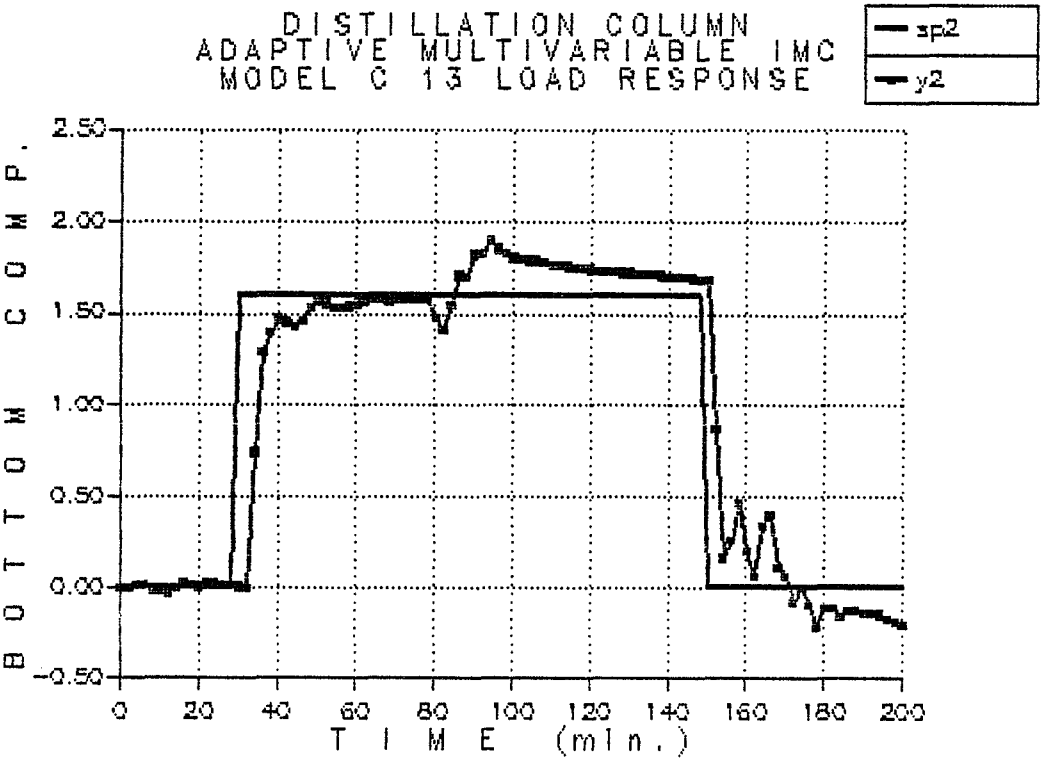
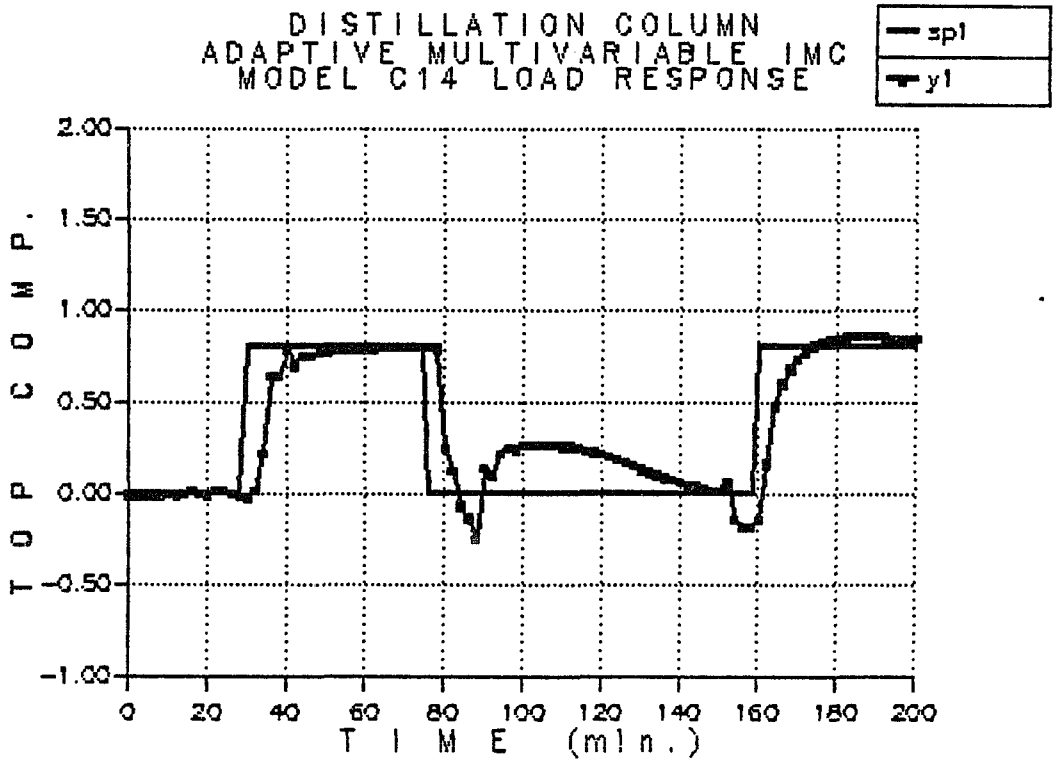

DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C 13 LOAD RESPONSE
sp2
y2
BOTTOM COMP.
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

Fig. G.2G


DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C14 LOAD RESPONSE
sp1
y1
TOP COMP.
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

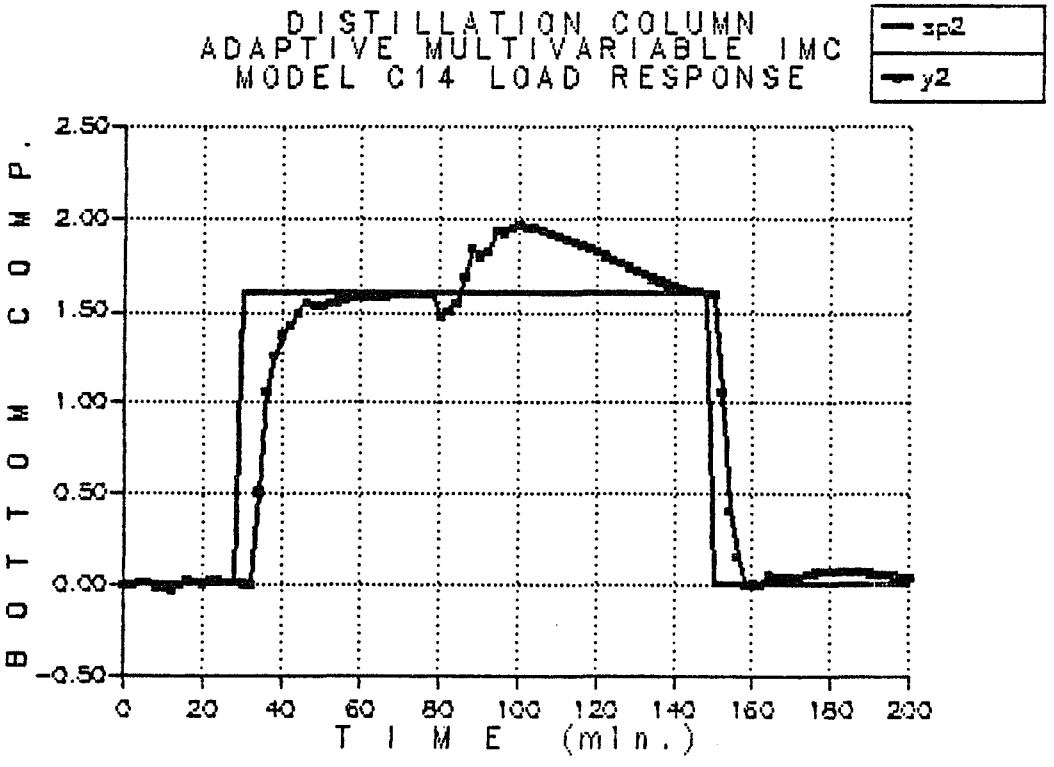
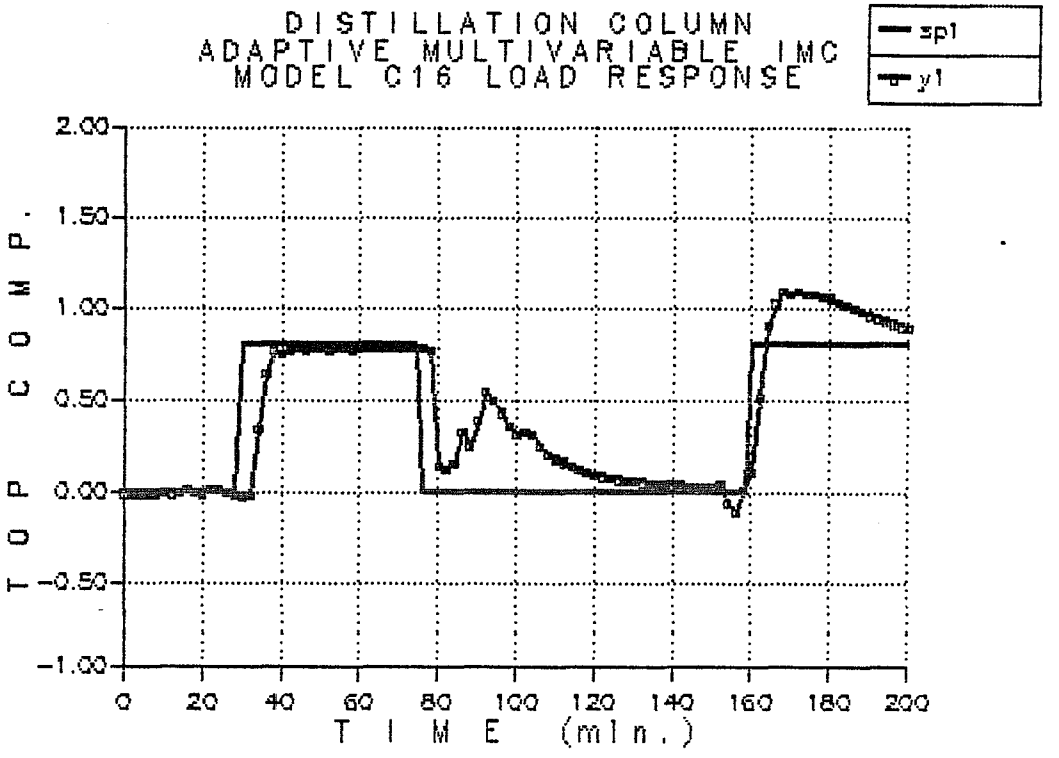

DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C14 LOAD RESPONSE
sp2
y2
BOTTOM COMP.
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

Fig. G.27


DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C16 LOAD RESPONSE
sp1
y1
TOP COMP.
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

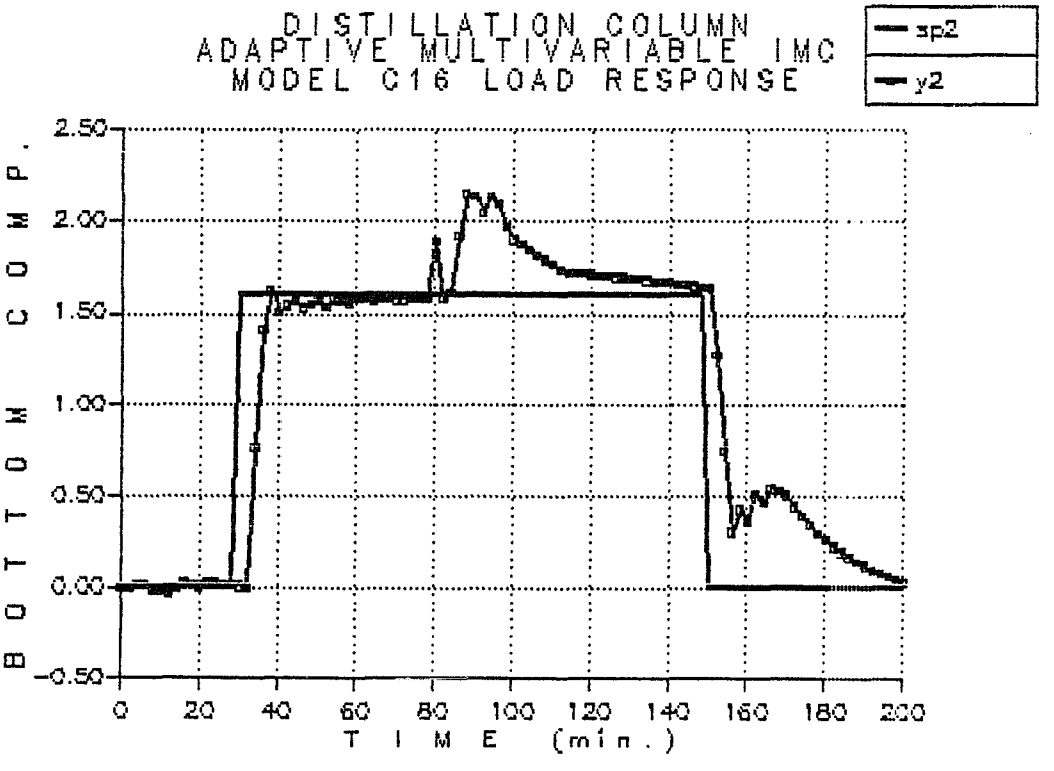
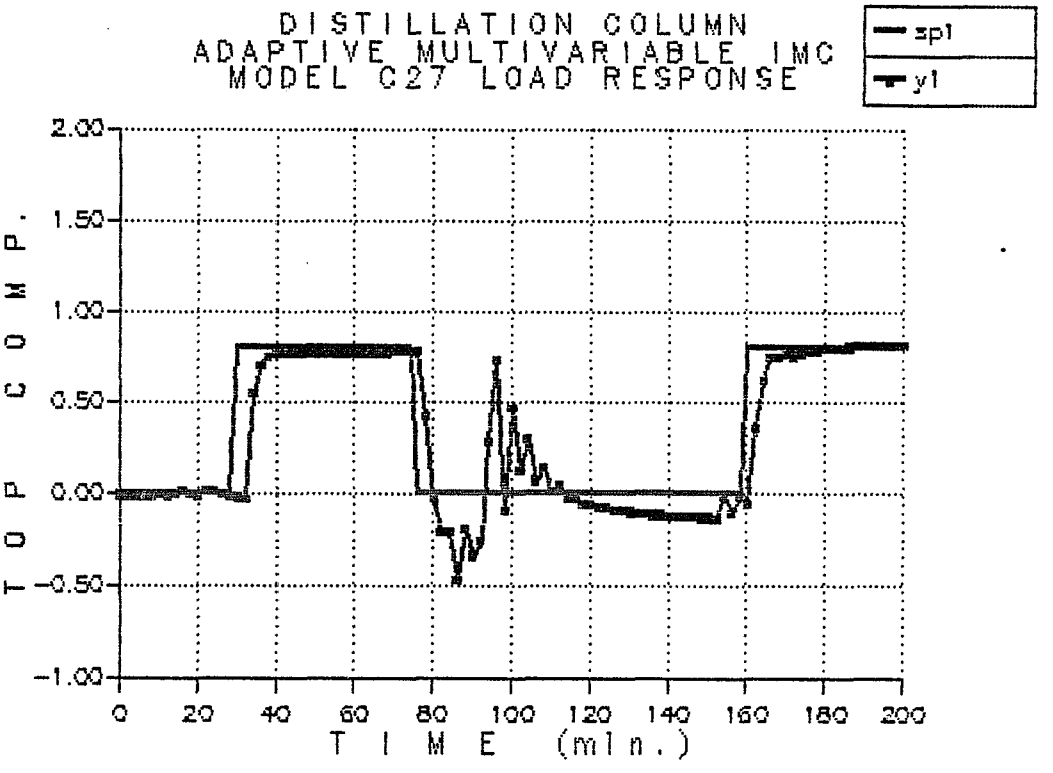

DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C16 LOAD RESPONSE
sp2
y2
BOTTOM COMP.
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

Fig. 6.28


DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C27 LOAD RESPONSE
sp1
y1
TOP COMP.
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

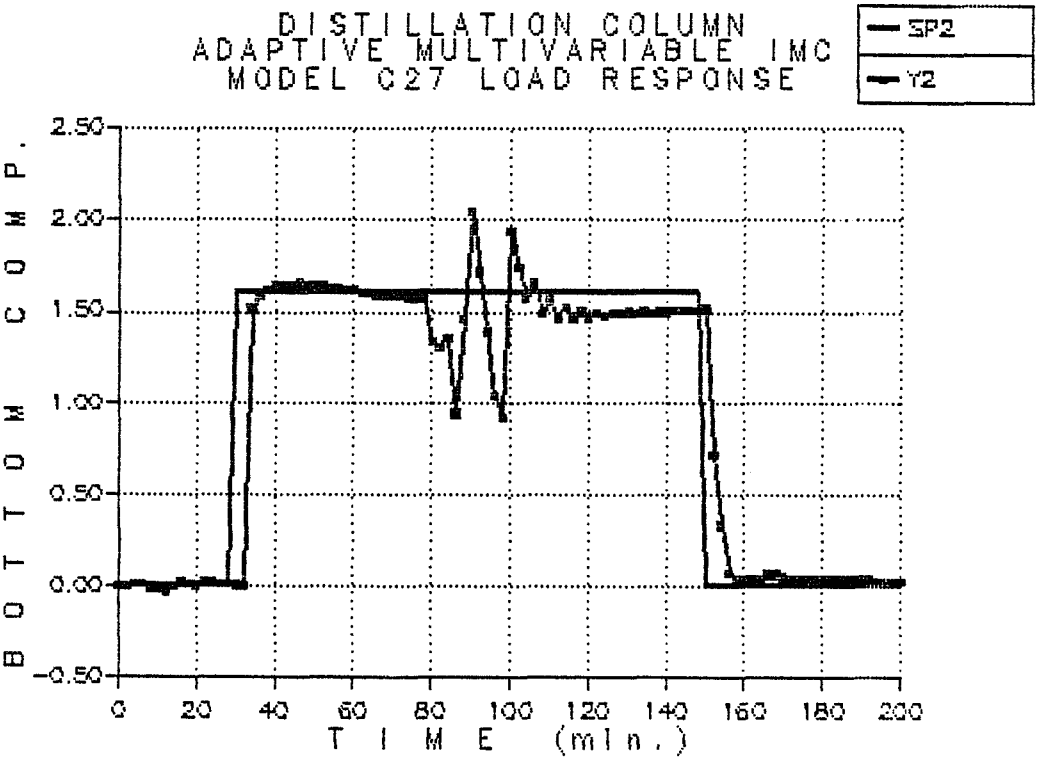

DISTILLATION COLUMN
ADAPTIVE MULTIVARIABLE IMC
MODEL C27 LOAD RESPONSE
SP2
Y2
BOTTOM COMP.
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
0
20
40
60
80
100
120
140
160
180
200
TIME (min.)

Fig. 6.29

time = 80 min. This load means, as it did before, that the plant characteristics change from system S1 to system S2. In this example, the robustness of the proposed control scheme can be observed with its capability to handle strong perturbations. It must be pointed out that whenever the information is persistent exciting, that is there exists sufficient information for determining the parameters, the adaptive internal model controller offers a great flexibility with respect to the model order. Four reasonable models show good performance under the same perturbation conditions.

6.3.3.- Non-linear chemical reactor

We can say that all physical systems are non-linear and time variant, but the most common way to represent them, is through linear and time invariant models. This is possible, and the models will describe adequately the process, as long as the non-linearities are mild and the deviations from the operating point are small.

Sometimes, nevertheless, the linear model is not able to predict in a correct form the behavior of a process. In this case, other representations, like bilinear forms or the use of non-linear estimation methods are necessary. Though more mathematics is involved, the non-linear models constitute a rich source of possibilities and the results justify their use. In order to set a boundary in the search of models and, for creating reasonable structures, a deeper in-

sight into the physical character of the non-linearities is necessary.

When dealing with symbolic mathematics i.e. block diagrams for linear or non-linear systems, our aim is to get the maximum knowledge about interconnected subsystems, starting from the knowledge of how each separated subsystem behaves. There are three elementary connections known as tandem, parallel and feedback. All other ways can be obtained as combinations of these elementary forms.

The most important properties of a system, i.e. stability and characterization are preserved through parallel connection. In case of tandem connection some problems arise with characterization of the whole system;some information may be lost due to pole-zero cancellation (for linear systems). The most complicated case, corresponds to the feedback connection where open loop characterization and stability, does not ensure these properties will be maintained when the loop is closed.(Indeed, a great part of the control efforts consist of how to design a subsystem so that it will produce a stable system when connected in a feedback path).

Stability may be established for linear invariant systems using Nyquist criterion, provided the process is SISO and known.Some extensions of this criterion have been developed for multivariable conventional systems (Arkun et al.(8)). The stability finally depends on the gains or am-

plification of the signals and phase delays while the information is flowing around the loop. Zames (96) successfully links the Nyquist criterion to the behavior of feedback establishing "that the stability of non-linear, time varying systems can often be assessed from certain gross features of input-output behavior, related with amplification and delay". Therefore, a general feedback system is closed loop stable provided the open loop gain is less than unity. As simple as is this criterion, though conservative sometimes, it constitutes a powerful design tool.

The above stability properties are general for any kind of feedback control. Besides, we must recall that the IMC structure is completely equivalent to conventional feedback control after redefining the control transfer function. Applications to internal model control structure.

The first step towards the design of a non-linear controller with internal model compensation corresponds to Economou et al.(26) who work with a single input single output known system.

The class of non-linear systems we are dealing with, is represented by the following operator equation:

$$\begin{aligned} x &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \qquad 6.6$$

where f, g and h belong to some fields on R . Economou et al (26) develop an inverse operator of (6.6) using a numerical

procedure based on the work of Hirschorn (40).

The developing of a right inverse operator is also studied by Nijmeijer (69) following a geometrical approach. Though the conditions for inverting the operator are important, in both cases, the algebraic and the geometrical approaches, the amount of mathematical manipulations is so cumbersome, that any other approach seems to be more desirable. Besides, a heavy knowledge of the structure and parameters of the system is absolutely necessary.(We must recall that the lack of adequate models is one of our problems). An alternative way of synthesizing non-linear controllers was explored by Kravaris and Chung (51). They apply the Lie derivative and brackets to get a linear system after using a non-linear transformation, the controller is designed over the linear system and then they come back to the original non-linear process. Once again, the necessary knowledge of the process and the mathematical complications make this method impractical.

IMC analysis.-

Returning to the internal model control structure, we can state that it is more important matching the model output to the plant output (opening the loop) than getting the exact inverse of the plant operator.

Assume P represents the non-linear plant and that G# is a linear model approximation such that $|| Pu - G\#u|| < c$ where u is the control vector and the norm is Euclidean.

Note that, up to this point, no requirements have been imposed on c, except being greater than zero. Later, a small c will mean a good modeling of the plant.

For a plant P the IMC structure gives:

$$u = (I - GcG\#)^{-1} Gc(sp - y).$$

So, if $Gc = G\#^{-1}$, the output y is independent of the given plant. That is, an infinite gain is achieved and the system will suffer from stability and sensitivity problems, unless c = 0, and that means the perfect modeling we know we cannot obtain.

The independence of y from G is true after the following:

For IMC systems

$$y = G(I + Gc(G - G\#))^{-1} Gc(sp - d) + d$$

assume the model is not accurate and G# is a bad representation of G.

Construct $Gc = G\#^{-1}$ then:

$$y = G(I + GcG - I)^{-1} Gc(sp - d) + d = sp$$

This is possible because the gain is infinite or $I - GcG\# = 0$. Since the model is not accurate, $y - y_m \neq 0$ and therefore the controlled process will be very sensitive or unstable.

A more simple approach under adaptive IMC structure

The basic aspect of IMC structure is to calculate a model output y_m such that it matches the plant y (in absence of load disturbances). An adaptive linear model can produce the appropriate matching output y_m nevertheless, the inverse operator for the linear model cannot match the inverse operator of the non-linear plant.By this reason, the synthesis of the "perfect" controller must be abandoned and a different way must be sought.

Let us study the following non-linear reactor presented by Tsiligiannis and Svoronos (86).

$$\dot{Ca} = F(Cao-Ca)/V-ko*exp(-E/RT)Ca$$

$$\dot{T} = F(To -T)/V+(-DH/\rho cp)ko*exp(-E/RT)Ca-UA(T-Tc)/V\rho cp$$

with

A : heat transfer area

Ca : reactant concentration

Cao : feed concentration

cp : specific heat

-DH : heat of reaction

E : activation energy

F : flow rate

ko : frequency factor

R : gas constant

T : reactor temperature

Tc : coolant temperature

To : reference temperature

U : overall heat transfer coefficient

ρ : density

Defining

x(1) = Ca/Cao

x(2) = T /To

u(1) = F /Fo

u(2) = Tc/To

we have

$$\dot{x}(1) = (1-x(1))*u(1)/Tau - kx(1)$$

$$\dot{x}(2) = (1-x(2))*u(1)/Tau + akx(1) - bx(2) + bu(2) \quad 6.7$$

$$k = ko*exp(-g/x(2)) \quad 6.8$$

$$a = -DHCao/V\rho cp$$

$$b = UA/V\rho cp$$

$$g = E/RTo$$

$$Tau = V/Fo$$

The measured outputs are:

y(1) = x(1)(t-2) dimensionless concentration

y(2) = x(2)(t) dimensionless temperature

Assuming the system is completely unknown, a first attempt was to represent the non-linear plant P by a linear model G# with $N_{AM} = 1$, $N_{BM} = 4$ and then construct Gc as the best inverse of G#. Though the values of c are very small

(but $\neq 0$), constructing $Gc = G\#^{-1}\ G_+$ means a high gain and therefore the controlled system is quite sensitive and prone to instability.

Figs. 6.30 and 6.31 show the best results for a step change of 0.5 units in the dimensionless concentration while the temperature set point remained constant. Although the filter F has been tuned with a strong damping effect, the results aren't satisfactory.

The offset is due to the fact that $G\#^{-1}(1)$ is not the true inverse of P(1). Even though c is very small and $\bar{d} \sim d$, the inverse of the linear model doesn't match the inverse plant operator. The high sensitivity is caused by the high gain controller.

Improving the non-linear IMC

The reactor model (6.7) belongs to a more restricted class of systems than those defined by (6.6). As a matter of fact, we can rewrite (6.7) as :

$$\dot{x} = \begin{bmatrix} -k & 0 \\ ak & -b \end{bmatrix} x + \begin{bmatrix} (1-x1)/Tau & 0 \\ (1-x2)/Tau & b \end{bmatrix} u \qquad 6.9$$

or

$$\dot{x} = A(x)x + B(x)u$$

$$y = h(x)$$

$x, y \in R^2$

Equation (6.9) is not a bilinear system, in spite of

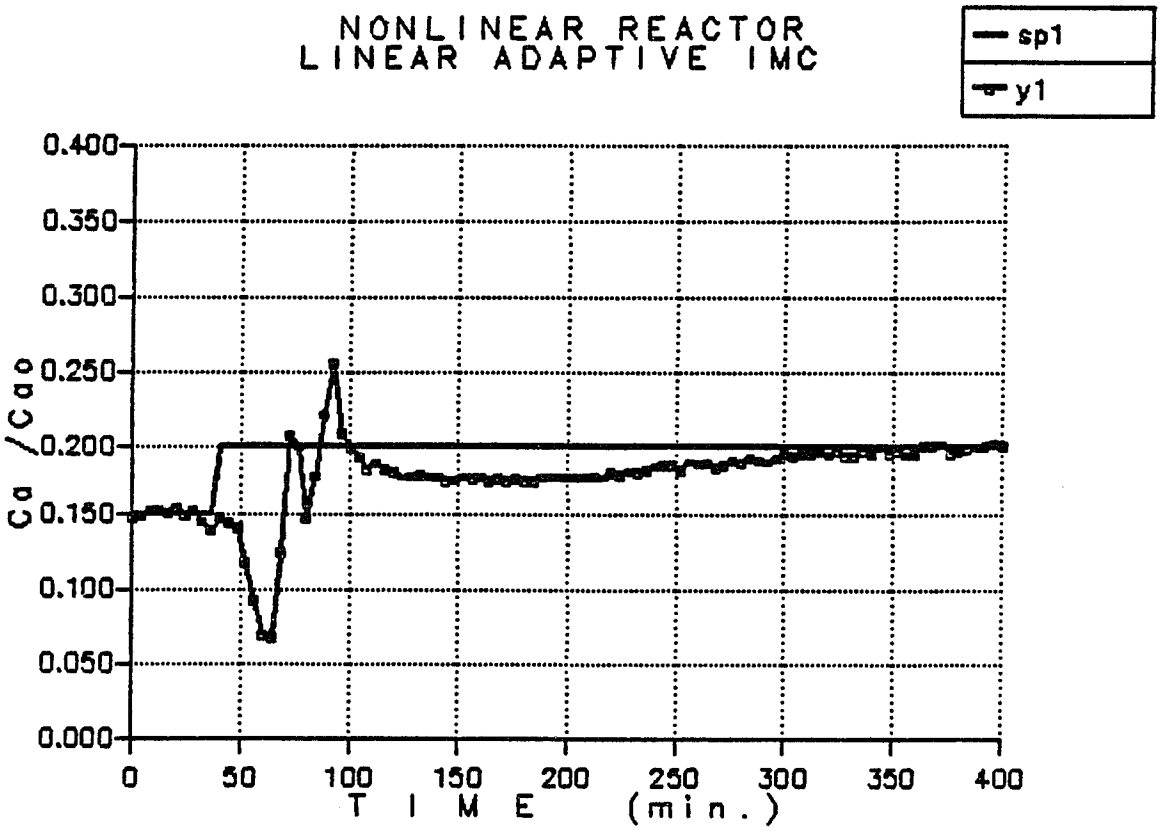

NONLINEAR REACTOR
LINEAR ADAPTIVE IMC
sp1
y1
0.400
0.350
0.300
0.250
0.200
0.150
0.100
0.050
0.000
Ca /Cao
0
50
100
150
200
250
300
350
400
T I M E (min.)

Fig. 6.30

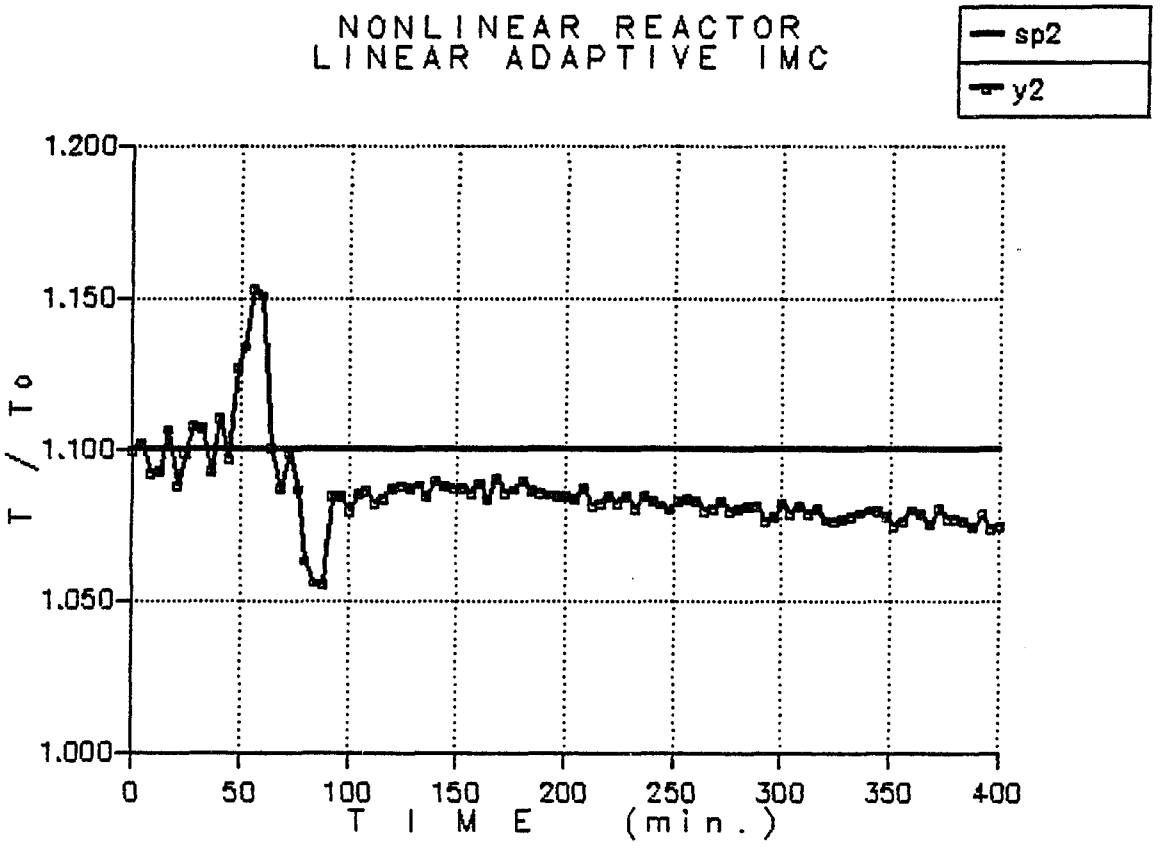

NONLINEAR REACTOR
LINEAR ADAPTIVE IMC
sp2
y2
1.200
1.150
1.100
1.050
1.000
T / To
0
50
100
150
200
250
300
350
400
T I M E (min.)

Fig. G.31

its aspect, due to the presence of the highly non-linear term k = ko*exp(-g/x(2))

At steady state, we can write:

$$x_s = -A^{-1}(1)*B(1)u_s = G_K u_s \qquad 6.10$$

G_K is not a transfer function matrix nor a differential operator but just a numerical gain matrix relating x_s and u_s. The argument 1 in matrices A and B means that their evaluation must be done at time = ∞ assuming for x(1) and x(2) the required set point values. In practice, this asymptotic behavior is obtained by evaluating G_K at each sampling time with the present set point values.

Having obtained the steady state gain matrix, Gc is designed as :

$$Gc = G_K^{-1} F \quad \text{where F is the IMC filter.}$$

In this way, we can estimate on line an adaptive model G# that makes it possible to follow the plant evolution for different operating points in such a way that $y_m \tilde{=} y$ and therefore the system behaves as open loop according to the IMC structure. This allows us to design Gc with the only restriction that it must be stable.

In Figs. 6.32 and 6.33 using the same model as before, we can appreciate the improved behavior of the system with respect to Figs 6.30 and 6.31.

Remark

The uncertainty about the parameters a, b, g and ko, can be eliminated by estimating them through the use of the same RLS algorithm with non-linear regressors (though the model itself is non-linear, ko is linear with respect to the regressor exp(-g/x(2)) and g is linear with respect to 1/x(2)). For more details about the non-linear estimation see Agarwal and Seborg (1, 2). It is not necessary to carry on this procedure on line because for a given system, the parameters are constant or almost constant. For example, g = E/RTo and E is known to experience small variations with temperature, but normally, the changes mean effects that are far beyond our accuracy measuring the temperature in an industrial process.

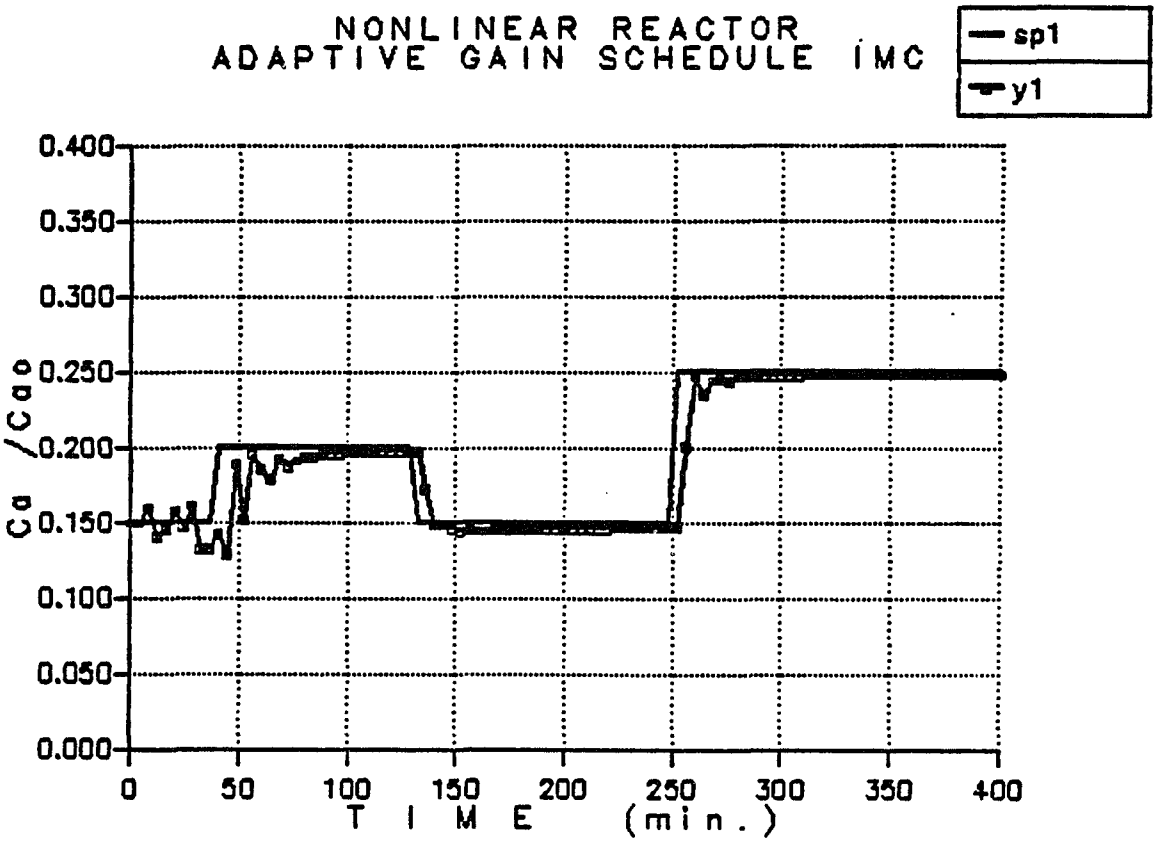

NONLINEAR REACTOR
ADAPTIVE GAIN SCHEDULE IMC
sp1
y1
0.400
0.350
0.300
0.250
0.200
0.150
0.100
0.050
0.000
Ca /Cao
0
50
100
150
200
250
300
350
400
T I M E (min.)

Fig. 6.32

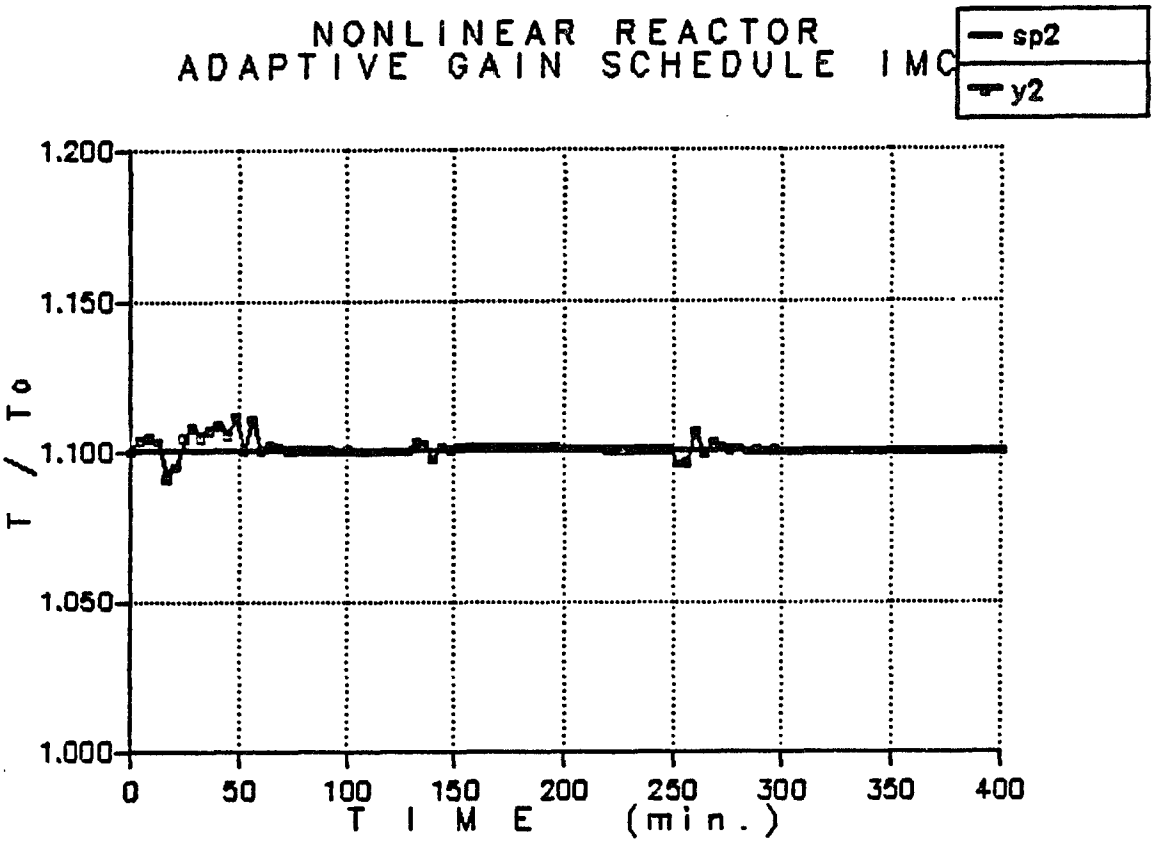

NONLINEAR REACTOR
ADAPTIVE GAIN SCHEDULE IMC
sp2
y2
1.200
1.150
1.100
1.050
1.000
T / To
0
50
100
150
200
250
300
350
400
T I M E (min.)

Fig. G.33

CHAPTER VII

DISCUSSION AND CONCLUSIONS

The performance of the proposed adaptive multivariable internal model control must be discussed from several points of view.

1.- Modeling

One of the points extensively discussed by Foss (30) is the difficulty of representing large multivariable processes by low order models. We can say that the phenomenological approach, that is the estimation of parameters without physicochemical meaning, fulfills the modeling needs and is completely adequate for control purposes. Furthermore, fixing a priori a matrix structure means transforming the identification problem into an estimation one with the order of the model as a unique parameter to specify. This scheme adds the necessary flexibility when dealing with large systems. In all the worked examples, the matching of the plant output y and the model output y_m was outstanding. We refer to matching the outputs instead of parameters because the "true" parameters do not exist for continuous systems and the estimated parameters for a discrete model are valid only for a given order with no other meaning as explained above. Another important reason is that $y - y_m$ has a very precise

meaning as the estimated load used as feedback signal by the internal model control structure.

With regard to computational characteristics, the multivariable estimation algorithm proved to be fast and reliable even with large systems with more than 50 parameters to estimate. Approximately 4 to 6 samples were enough to estimate 8 parameters (Fig.6.2) and 20 to 25 samples for estimating 36 parameters in model C27 for the distillation column. For stable processes, the estimated parameters always converge to the "true parameters" in absence of noise. Theoretically, the expected values of the estimated parameters converge to the true parameters when the observed signal is corrupted with white noise.

2.- Inversion and control synthesis

The polynomial matrix inversion algorithm worked on line without difficulties and doesn't present numerical problems even with ill-conditioned matrices. This characteristic results as a consequence of the procedure where the adjoint matrix and the determinant are calculated separately. The adjoint matrix always exists independent of the values of det(B). In the worst case, det(B) can be identically zero. Obviously, further calculations cannot be pursued and the simulation is interrupted at this stage.

Regarding computational time, this algorithm is slower than the estimation algorithm. In order to save time, redun-

dant calculations of the inverse and Gc were avoided. If the estimation error was smaller than a given threshold (that is, the model was not being updated or adapted) then, the inversion step was bypassed and Gc was held equal to the last calculated value. This scheme was specially remarkable with the distillation column example. The simulation was faster with model C27 (36 parameters) than with model C14 (20 parameters). The explanation is the accuracy of model C27 so that for a given threshold, the inversion subroutine was called more often by the system operating with the model C14.

Estimated computational times are of order 0.05 seconds per sample.

3.- Controlled systems

-Paper machine headbox.

This system was presented by Astrom in a continuous version. Borison (15) presents the discrete model and his work is oriented to stochastic properties over a non varying plant.

In our results, the self-tuning characteristics are explored first. Then the adaptive capabilities are tested introducing changes in the plant parameters each time the operating point is changed. At this stage, when a large amount of estimation error is produced, we tried to back up the IMC with a conventional three modes controller but the tuning difficulties made evident that the adaptive IMC was a

better alternative.

The control difficulties offered by a changing plant are emphasized by the behavior of the non adaptive IMC, Figs. 6.13 and 6.14.

-Distillation column

The continuous model was presented by Wood and Berry (92) after fitting experimental data from a pilot unit. In their work, these authors show the severe interaction that exists for a conventional two points control system. Their results show a deterioration of performance, tuning difficulties and an extremely slow and oscillatory response. They improved the system by introducing a compensatory decoupling scheme but they conclude that it is very difficult to find the adequate compensator without a good model of the process.

Model S1 (6.4), developed by Wood and Berry, was used in a simulation study by Ogunnaike and Ray (71). They introduced the multiple delays compensator, a kind of generalized Smith predictor, but they must assume the plant is perfectly known and invariant.

Our results for the self-tuning internal control are better than those presented by Ogunnaike and Ray, and equivalent to the theoretical results presented by Garcia and Morari (32) for the same system.(We mean by theoretical results those provided not through simulation but directly from the IMC expression $y = G_+F(sp-d)+d$).

Arulalan and Deshpande (9) present a simplified predictive control synthesis approach using IMC structure and the same column model S1. All our results are advantageously compared with those obtained using the predictive method, showing better set point tracking properties, no overshoot and no oscillations.

The importance of persistent exciting signals must be remarked once more. The quality of system identification (parameters estimation) and henceforth the control action are improved whenever sufficient information is provided to the estimation algorithm. This statement is critically true when the plant is being modified by the entering perturbations (see Figs.6.24 to 6.27).

-Non-linear reactor

Whenever the process presents a strong non-linear behavior, beyond the sporadic change of parameters, the fast parameter adaptation makes it highly appropriate for internal model control structure. Nevertheless, the control function, calculated as the inverse of the linear approximate model, leads to a highly sensitive controller with additional offset problems. In this case, the controller can be calculated on-line as an adaptive gain schedule with a minimum number of parameters, nevertheless these parameters are calculated off-line.

Tsiligiannis and Svoronos (86) present a reactor system

with different delays affecting the outputs. The control strategy is ill defined with a low operating range and low sensitivity for one of the manipulated variables, (inlet temperature). The second control variable is the reactant feed flow. They assume the delays are known.

The adaptive IMC presented comparable or better results under more restrictive and realistic conditions for the inlet temperature.

An alternative way for estimating parameters in this example consists in the use of deviations variables in the estimation algorithm. (suggestion made by Dr. A.B.Corripio, personal communication).

4.- Load disturbances

Sustained or changing deterministic loads pose a serious problem to adaptive systems. Most of the self-tuning regulators presented in literature are designed in terms of stochastics signals and only a few authors such as Clarke (21) and Fortescue (29) refer to these difficulties. According to Clarke, the method of "one in the regressor vector" is adequate to sustained load but cannot cope with fast varying signals. Our results show that it is possible to estimate a sustained load and use this estimation as a feedback signal, according to IMC algorithm. We must remark that the load was not an isolated perturbation but also introduced strong changes in the process parameters. However, the

adaptive IMC was able to overcome these problems.

5.- Multiple dead times

Until the work of Smith, the presence of dead time posed a problem to control design. Alevisakis at al.(3, 4) extended the Smith predictor to multivariable delays but their method can only be applied to systems with the same delay in all variables. Ogunnaike and Ray (71) generalize the Smith pre- dictor to multiple delays but their method requires a perfect knowledge of the system.

From the adaptive point of view, Tsiligiannis and Svoronos (86) present a reactor system with multiple delays but they must be known in advance. In our proposed adaptive multivariable internal model control, we followed the methodology suggested by Vogel (88) and we do nothing in advance with respect to dead times. Through the explicit parameters estimation, the eventual dead times in the model will appear naturally according to the procedure described in (5.22). The different dead times are then compensated in a decoupled way (algorithm BMIN). As discussed by Morari (66) if the dominant dead times cannot be located in the main diagonal of G through permutations of variables, then the compensation is not optimal and can be improved with a scheme that allows interaction. In our work we used a non interacting procedure because we estimate that total decoupling is more important than a few samples of delay in the response. From

the computational point of view, the decoupled suboptimal case is far more simple than an optimal algorithm.

CONCLUSIONS

A new adaptive multivariable control scheme has been devised. The method combines the best characteristics of conventional adaptive systems and internal model control structure, i.e.capability of working with explicitly unknown or almost unknown processes. The necessary modeling is internally achieved without the negative properties that usually make the control design unclear or difficult such as specifying weighting matrices, poles placement, pairing variables nor problems associated with stability considerations. The outstanding characteristics of internal model control are used and the control algorithm proceeds much as a conventional IMC where the plant model is always accurate and well known in spite of sudden modifications that can affect the process.

The programs were successfully implemented through the following stages:

1.- Identification

A fixed matrix structure was adopted using the input and output orders of the model as degree of freedom.

For simplicity's sake, the well known recursive least

square algorithm with variable forgetting factor was adopted. The extension to the multivariable case proved to work smoothly and free of problems even with a large number of parameters to be estimated.

2.- Control

Internal model control structure is based on the appropriate modeling and inversion of the model. For multivariable processes this inversion means working with polynomial matrices. At the same time, the inherent control difficulties associated with the plant dynamics must be solved through the factorization of the polynomial matrix. In this work, an alternative path was found, consisting of on-line matrix inversion and then factorization. Using the McMillan form properties but without calculating it explicitly, the matrix factorization was reduced to a scalar polynomial factorization.

3.- Filter parameters

The usual non adaptive IMC filter was adopted as a way to attenuate the sensitivity of the controller. No innovations were implemented and the simplest diagonal matrix with first order lags was used. The attenuation parameters were used as tuning parameters.

The simulation results show that the new scheme presents excellent self-tuning properties for fixed systems, adaptive

capacity for time variant or non-linear processes, good set point tracking properties, robustness for rejecting sustained loads, insensitivity to model order, insensitivity to unknown and varying dead times and no stability considerations are necessary.

Finally, we must point out that one of our simulation with a non adaptive IMC system shows that the BIBO stability condition by itself does not assure performance and must be used cautiously.

SUGGESTED WORK TO BE DONE

1.- Extension of the controller synthesis to the general (mxn) case

Whenever the number of inputs is greater than the number of outputs, this case may be reduced to the (mxm) case without loss of generality. We can use the degree of freedom to choose the best possible set of manipulated variables and then delete the m-n that appear less appropriate.

If the outputs exceed the inputs then the problem is far more complex because the rank of matrix B must be at least equal to the number of inputs m. Instead of the generalized Penrose inverse, some adaptive gain schedule may be preferred.

2.-To extend the study of non sustained load problem to the MIMO case

So far, the load problem with adaptive system has

barely been studied for multivariable systems. Perhaps the best available procedure is presented by McDermott and Mellichamp (62) but their algorithm has an overwhelming complexity.

3.- Application of the inversion algorithm as a computer aided design system

Conventional multivariable design procedures require the inversion of a known model in order to synthesize the appropriate controller. Whenever this model is large, the task to produce a feasible inverse is not an easy one. At this point, the inversion algorithm can construct the best approximate inverse (from the decoupling point of view).

4.- To extend the adaptive algorithm to work through an interface with ACS. The advanced control system (ACS) has many characteristics that makes attractive the possibility of connecting an adaptive algorithm.

5.- To apply the adaptive IMC to a real process by using a microcomputer. (Appendix A).

CHAPTER VIII

REFERENCES

1.- Agarwal M.,Seborg D."Self tuning controllers for non linear systems".
Automatica,V.22 # 2,pp 209-214,1987.

2.- Agarwal M.,Seborg D."A multivariable nonlinear self tuning controller".
AIChE J.,V 33 #8 pp 1379-1386,1987.

3.- Alevisakis G.,D.E.Seborg."An extension of the Smith predictor method to multivariable linear systems containing time delays".
Int. J.of Control,V 17 N 3 pp541-551,1973.

4.- Alevisakis G.,D.E.Seborg."Control of multivariable systems containing time delays,using a MV. Smith predic tor".
Chem.Eng.Science,V 29 #2 pp 373-380,1974.

5.- Allidina A.,F.Hughes."A general adaptive scheme".
Int.J.of Control,V 38 #6 pp 1115-1120,1983.

6.- Amaral W.,L.Magalhaes,M.Mendez."Implementation of a self tuning regulator in a minicomputer"
ISA J.pp 221-228,1978.

7.- Anderson B.D.,C.R. Johnson."Exponential convergence of adaptive identification and control algorithms".
Automatica,V 18 pp 1-13,1982.

8.- Arkum Y. et al."Computer aided analysis and design of robust multivariable control systems for chemical process".

Computers & Chem.Eng., V 9 #1 pp 27 59, 1985

9.- Arulalan G., Deshpande P."Simplified model predictive control".

Ind.Ing.Chem.Res., V 26 pp 347-356, 1987

10.- Astrom K., U.Borison, L. Ljung. "Theory and application of a self-tuning regulator".

6th IFAC Congress on Control technology.Cambridge Mass.1973.

11.- Astrom K., B.Wittenmark."On self-tuning regulator".

Automatica, V 9 pp 185-199, 1973.

12.- Astrom K.Computer Controlled Systems.

Prentice Hall Inc.Eng.Cliff, 1984.

13.- Bart H et al."Factorization of transfer functions".

SIAM J.Control and Optimization.V 18 # 6 pp 675-696, 1980.

14.- Bierman G.J. Factorization Methods for Discrete Sequential Estimation.

Academic Press, N.Y., 1977.

15.- Borison U."Self tuning regulator for a class of multivariable systems".

Automatica, V 15 pp 209-215, 1979.

16.- Buslowicz M."Inversion of polynomial matrices".

Int.J.of Control, V 33 # 5 pp 977-984, 1980.

17.- Chen T.S. Linear System Theory and Design.
Holt, Rinehart and Winston.N.Y., 1984

18.- Chien I.,D.Seborg, D.Mellichamp."A self-tuning controller for systems with unknown or varying time delay".
IFAC pp 905-910, 1986.

19.- Chien I., D.Seborg, D.Mellichamp."Self tuning control with decoupling".
AIChE J. V 33 # 7 pp 1079-1088, 1987.

20.- Clarke D., P, Gawthrop."Self tuning control".
Proc.IEE. V 122 # 9 pp 929-934, 1975.

21.- Clarke D.Introduction to Self Tuning Controllers
IEEE Cont.Eng.Series 15, Peter Peregrinus Ltd.
London 1985.

22.- Clarke D., et al."A general LQG application to self tuning control.Part I".
Int.J.of Control, V 41 # 6 pp 1509, 1523, 1985.

23.- Clarke D.et al."A general LQG application to self tuning control.Part II".
Int.J.of Control, V 41 # 6 pp 1525-1544, 1985.

24.- Corripio A.B., P.M.Tompkins." Industrial applications of a self-tuning feedback control algorithm".
ISA Trans., V 20 # 2, pp 3-10, 1981.

25.- Dahlin E.B."Designing and tuning digital controllers".
Instr.Control Syst., V 41 # 6, pp 77- 83, 1968.

26.- Economou G., Morari M."Internal model control 5; Extension to non-linear systems".

Ind.Eng.Chem. Proc.Dis & Develop. V 25 pp 403-411,1986.

27.- Egardt B."Unification of some discrete-time adaptive schemes".
IEEE Trans.AC-25 pp 693-697, 1980.

28.- Evans R.J., G.C.Goodwin and R.E.Betz.Discrete Time Adaptive Control for Classes of Nonlinear Systems. Information and Optimization.Springer, 1981.

29.- Fortescue T.et al." Implementation of Self tuning regulators with variable forgetting factor".
Automatica V 13 pp 831-836 , 1981.

30.- Foss A.S."Critique of chem.process control".
AIChE J. V 19 pp 209-214, 1973.

31.- Garcia C., M.Morari." Internal model control 1".
Ind.Eng.Chem. Proc.Des.& Develop. V 21 pp 308-323,1982.

32.- Garcia C., M.Morari." Internal model control 2".
Ind.Eng.Chem. Proc.Des.& Develop. V 24 pp 472-484,1985.

33.- Garcia C., M.Morari." Internal model control 3".
Ind.Eng.Chem. Proc.Des.& Develop. V 24 pp 484-494,1985.

34.- Gomart G., P, Caines. "Robust control of time varying systems".
Proc.of the 23th conference on decision and control, pp 1021- 1024, 1984.

35.- Goodwin G.C. et al."Discrete time multivariable adative control".
IEEE Trans.AC-25 pp 449-456, 1980.

36.- Goodwin G.C., C.R.Johnson and K.S.Sin."Global conver-

gence for one step ahead optimal controllers based on input matching".

IEEE Trans.AC-26 pp 1269-1273, 1981.

37.- Goodwin G.C., K.S.Sin.Adaptive Filtering Prediction and Control.

Prentice Hall Inc.Englewood Cliff, 1984

38.- Grimble G."Implicit and explicit LQG Self tuning control ".

Automatica V 20 pp 661-669, 1984.

39.- Hiram Y., L.Kershenbaum."A report on simulation studies of extended horizon self-tuning regulator".

I.College.1983.

40.- Hischorn D."Output tracking in multivariable non-linear systems".

IEEE Trans AC-26 pp 593-595, 1981.

41.- Holt B.,M.Morari."Design of resilient processing plants V".

Chem.Eng.Sci. V 40 #7 pp 1229-1237, 1985.

42.- Householder A. The Theory of Matrices in Numerical Analysis.

Dover Publications Inc.N.Y., 1964.

43.- Inouye V."An algorithm for inverting polynomial matrices".

Int.J.of Control, V 30 pp 989-999, 1979.

44.- Joseph B., C.B.Brosilow."Inferential Control of Processes.Parts I, II and III".

AIChE J. V 24 # 3 pp 485-509, 1978.

45.- Kalman R."Design of a self optimizing control system". Trans.ASME pp 468-478, 1958.

46.- Kantor J., R.Andres."Characterization of allowable perturbation for robust stability". IEEE Trans.AC-28, # 1 pp 107-109, 1983.

47.- Kantor J."The analysis of robust stability and performance in multivariable feedback systems using M-matrices". IFAC pp 1309-1314, 1985.

48.- Kestenbaum A. et al."Design concepts for process control". Ind.Eng.Chem.Proc.Des.and Develop. V 15 pp 1-12, 1976.

49.- Koivo H."A multivariable self-tuning controller". Automatica V 16 pp 351-366, 1980.

50.- Kontos A."An algorithm to factor polynomial matrices". Technical Report # 7914.Rice Univ., 1979.

51.- Kravaris C., Chung C."Nonlinear steady state feedback synthesis by global input-output linearization". Proc.J.Amer.Cont.Conf. pp 997-1005, 1986.

52.- Landau Y.D.Adaptive Control. The Model Reference Approach. Marcel Dekker N.Y., 1979.

53.- Lang S., X.Gu, T Chai. "A multivariable general self-tuning control with decoupling design". IEEE Trans.AC-31 pp 474-477, 1986.

54.- Lauhghlin D. et al."Internal model control and process uncertainty".

Int.J.Cont. v 44 # 6 pp 1675-1698, 1986.

55.- Lee K., Lee W."Extended discrete time MV adaptive control using long term predictor".

Int.J.of Control, V 38 # 3 pp 495-514, 1983.

56.- Lee W., V.Weekman Jr."Advanced control practice in the chemical industry".

AIChE J. V 23 # 1 pp 27-38, 1976.

57.- Levien K., M.Morari." Internal model control of coupled distillation columns"..

AIChE J. V 33 # 1 pp 83-98, 1987.

58.- Lindorf D., R. Carroll."Survey of adaptive control using Liapunov design".

Int.J.of Control, V 18 # 5 pp 897-914, 1973.

59.- Ljung L. Systems Identification.Theory for the User. Prentice Hall Inc.Inf.and Systems Sc.Serie.N.J., 1987.

60.- Manousiouthakis V., Y.Arkun." Hybrid approach for the design of robust control systems".

Int.J.of Control, V 45 # 6 pp 2203-2220, 1987.

61.- McDermott P., D.Mellichamp."An autopole placement self tuning control".

Int.J.of Control, V 40 # 6 pp 1131-1147, 1984.

62.- McDermott P.,D.Mellichamp."A decoupling pole placement self-tuning control for a class of multivariable systems".

Optimal Control Applic.& Methods.V 7 pp 55-79,1986.

63.- Mohammadi N.A.,G.L.Rempel " Control,data adquisition and analysis of catalitic gas-liquid reactors using a personal computer".
Comput.Chem.Engng.V 11 # 1 pp 27-35,1987.

64.- Monopoli R.V."Model reference adaptive control with augmented error signal".
IEEE Trans.AC-19,pp 474-484,1974.

65.- Moore B."On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignation".
IEEE Trans.AC-21 # 5 pp 689-692,1976.

66.- Morari M."Design of resilient processing plants IV".
Chem.Eng.Sci. V 38 # 11 pp 1881-1891,1983.

67.- Morris A.,Nazer Y.,R.K.Wood."Single and multivariable applications of self-tuning control".
Self tuning and Adaptive Control.C.J.Harris And S.A. Billings Ed.IEE Control Eng.Series. 15,1981.

68.- Newell R.B,D.G Fisher."Model development,reduction,and experimental evaluation for an evaporator".
Ind.Eng.Chem.Proc.Des.and Develop. V 11 # 2 pp 213-221, 1972.

69.- Nijmeijer H."Right invertibility for a class of non-linear control system:A geometric approach".
Systems and Control Letters.V 17 pp 125-132,1986.

70.- Ogunnaike B."An optimal controller for discrete time

delay systems requiring no prediction".
Chem.Eng.Commun. V 37 pp 249-264,1985.

71.- Ogunnaike,B,W.Ray."MV controller design for linear systems having multiple time delays".
AIChE J. V 25 # 6 pp 1443-1056,1979.

72.- Palazoglu A.,Y.Arkum." Robust tuning of process control systems using singular values and their sensitivities".
Chem.Eng.Commun. V 37 pp 315-331,1985

72b.-Patel R. V.,N. Munro. Multivariable system Theory and Design. Pergamon Press,N.Y. 1982.

73.- Pernebo L."An algebraic theory for the design of controller for linear multivariable systems".
IEEE Trans.AC-26 # 1 pp 171-194,1981.

74.- Peterka V.,K.Astrom."Control of multivariable systems with unknown but constant parameters".
IFAC pp 535-544,1973.

75.- Peterka V."Adaptive digital regulation of noisy systems".
IFAC Symposium on Identification and Parameters Estim. 1970.

76.- Peterka V."A square root filter for real time multivariate regression".
Kybernetika V 11 # 1 pp 54-66,1975.

77.- Porte B.,M.Tatnall."Performance characteristics of MV MR adaptive systems synthesized by Liapunov method".
Int.J.Cont. V 10 # 3 pp 241-257,1969.

78.- Rohors C.L. et al."Robustness of adaptive control algorithms in the presence of unmodelled dynamics" IEEE Trans.AC-30,V 9 pp 881-889,1985.

79.- Rotea M.,Machetti J."Internal model control using the linear quadratic regulator theory". Ind. Eng.Chem. Res. V.26,pp 577-581,1987.

80.- Seborg D.,et al."Adaptive control strategies for process control :A survey". AIChE Diamond Jubilee Meeting pp 1-132,1983.

81.- Seborg D.,et al."Adaptive control strategies for process control :A survey". AIChE J. V 32,# 6,pp 881-913,1986.

82.- Smith J.M."Closer control of loops with dead time" Chem.Eng.Prog. V 53 # 5 pp 217-219,1957.

83.- Svoronos S."An adaptive internal model control". Proc.Amer.Cont.Conf. pp 1560-1565,1985.

84.- Svoronos S."Linear model dependent control". AIChE J. V 33 # 3 pp 394-400,1987.

85.- Tsiligiannis C.,S.Svoronos." Multivariable self-tuning control applicable to systems with measurements of differing time delays". Proc.Am.Cont.Conf. pp 899-904,1985.

86.- Touchstone A.T.,A.B.Corripio."Adaptive control through instrumental variable estimation of discrete model model parameters". Instrumentation in Chemical and Petroleum Industries.

ISA,Anaheim,CA,pp 1-10,1977.

87.- Vidyasagar M. Control Systems Synthesis.A Factorization Approach.
The MIT Press Ca.Mass.,1985.

88.- Vogel E.,T.Edgar."Application of an adaptive pole-zero placement controller to chemical process with variable dead time".
Proc.Am.Cont.Conf. pp 536-544,1983.

89.- Wellstead P.E,D.L.Prager."Self tuning multivariable regulator".
Self tuning and Adaptive Control.C.J.Harris And S.A. Billings Ed.IEE Control Eng.Series. 15,1981.

90.- Wittenmark B."A self-tuning predictor".
IEEE Trans.AC-19 # 6 pp 848-851,1974.

91.- Wolovich W.A.,P.L.Falb."Invariants and canonical forms under dynamic compensation".
SIAM J. Cont.Opt. v 14 pp 996- ,1976

92.- Wood R.K.,M.W. Berry."Terminal composition control of a binary distillation column".
Chem.Eng.Sci. V 28 pp 1707-1717,1973.

93.- Xianya X.,R.Evans."Discrete adaptive control for deterministic time varying systems".
Automatica V 30 # 3,pp 309-319,1984.

94.- Ydstie B.,et al."Theory and application of and extended horizon STC".
AIChE J. V 31 # 11 pp 1771-1780,1985.

95.- Zafiriou E.,M.Morari."Design of the IMC filter by using the structured singular value approach".
Proc.Am.Cont.Conf.pp 1-6,1986.

96.- Zames G."On the input-output stability of time variant non-linear feed back systems.I and II".
IEEE Trans.AC-11 # 2 pp 228-238 & 465-476,1966.

APPENDIX A

POSSIBILITY OF USING A MICROCOMPUTER

One of the most attractive characteristic of adaptive control schemes is the possibility of using small and cheap microcomputers integrated to the plant itself.

Nevertheless, the use of microcomputers is not free of problems mainly due to the short length of computer word (8 or 16 bit) that increases the rounding errors. This problem, related to adaptive systems is studied by Amaral et al.(6) and they give special emphasis to the numerical difficulties associated with the estimation algorithm. In order to avoid further problems with the numerical stability of the information matrix P, they suggest replacing this matrix by its triangular square root so that $P = QQ^T$. This procedure corresponds to Peterka's algorithm (76) up on which the estimation is performed in our work.

In their work, Amaral et al.used a PDP-11/05 computer with floating point, to implement the adaptive controller. They report in the slower case 39 ms.per step for the estimation algorithm. Similar performance is reported using a INTEL 8080 microprocessor.

At present time, the INTEL 8080 is an old fashioned piece of equipment that has been replaced by the 80386 family, three to five times faster. In the same way, the use

of a math co-processor can improve the calculation of some functions in an order of magnitude of two.

Computer language

The use of a high level computer language as FORTRAN, is attractive due to the similarities with current speech. Nevertheless, these languages make the performance of microcomputers slow and particularly inefficient for real time applications, and it is convenient using an intermediate language that combines the good characteristics of FORTRAN and the speed of low level languages.

In this sense, C programming language is well suited to real time applications because is specially designed for control purposes with many of the outstanding characteristics of PASCAL or BASIC and with direct access to memory locations. C compilers are available for most 8 bit, 16 bit and 32 bit microprocessors and many microcomputers.

Mohammadi and Rempel (63) work with a IBM personal computer with modest memory characteristics to control, data adquisition and analysis of a catalytic pilot reactor, using BASIC for programming the control software.

With the access to more and more powerful microcomputers, using 32 bit word length and running up to 20 Mhz, the future applications of adaptive software running with microcomputers specially assigned to specific process units seems to be a real possibility.

COMPUTER PROGRAMS

```
C
      SUBROUTINE REFIL(ERR,AB,N,M,FI,TIME)
C
C ***************************************************************
C *                                                             *
C * REFIL ROUTINE CALCULATES THE PARAMETERS OF A DISCRETE       *
C * MODEL.FI IS THE REGRESSOR VECTOR,M IS THE NUMBER OF         *
C * COLUMNS IN MATRIX AB AN N IS THE NUMBER OF VARIABLES.       *
C * THE PARAMETERS ARE STORED IN MATRIX AB WITH NAM             *
C * COLUMNS BELONGING TO MATRIX A AND NBM TO MATRIX B.          *
C * COVI IS THE INITIAL VALUE FOR THE INFORMATION MATRIX        *
C * SIGMA IS THE WINDOW PARAMETER                               *
C * R IS THE NOISE VARIANCE.                                    *
C *                                                             *
C ***************************************************************
      REAL GAIN(20),AB(5,20),SQCOV(650),FI(50),ERR(5)
      COMMON/FIVE/ ERROR
      DATA SQCOV/650*0./,GAIN/20*0./
C     DATA COVI/1.E14/,SIGMA/0.0020/,EPS/0.000001/,R/1.00/
      DATA COVI/1.E 8/,SIGMA/0.0500/,EPS/0.000001/,R/0.01/
      IF(TIME.EQ.0.) THEN
      I10      = 0
      TO       = M*COVI
      TRP      = TO
      SQCOVI   = SQRT(COVI)
      DO 40 I  = 1,M
      I10      = I10+I
      SQCOV(I10) = SQCOVI
   40 CONTINUE
      ENDIF
      I10      = 0
      I50      = 0
      I80      = 0
      W        = 0.
      DO 10 J  = 1,M
      FJ       = 0.
      DO 5  I  = 1,J
      I10      = I10+1
      FJ       = FJ +SQCOV(I10)*FI(I)
    5 CONTINUE
      W        = W+FJ*FJ
   10 CONTINUE
      ERROR    = 0.
      DO 20 L  = 1,N
   20 ERROR    = ERROR+ERR(L)*ERR(L)
      RK       = 1.-W-ERROR/SIGMA
      V        = RK*RK+4.*W
      ALFA     = (SQRT(V)+RK)/2.
      ALFMIN   = TRP/TO
      IF(ALFA.LT.ALFMIN) ALFA = ALFMIN
      IF(ALFA.GT.1.) ALFA = 1.
      PHI      = SQRT(ALFA)
```

```
      SIGMAJ  = PHI
      DO 100 J= 1,M
      FJ      = 0.
      DO 50 I = 1,J
      I50     = I50+1
      FJ      = FJ+SQCOV(I50)*FI(I)
   50 CONTINUE
      AUXA.sk0;C  . = SIGMAJ/PHI
      SIGM2J  = SIGMAJ*SIGMAJ
      AUXB    = FJ/SIGM2J
      SIGM2J  = SIGM2J+FJ*FJ
      SIGMAJ  = SQRT(SIGM2J)
      AUXA.sk0;C   = AUXAC/SIGMAJ
      GAIN(J) = SQCOV(I50)*FJ
      SQCOV(I50) = AUXAC*SQCOV(I50)
      I80     = I80+1
C     IF(J.EQ.1) GO TO 80
      DO 80 I = 1,J-1
      AUXD    = SQCOV(I80)
      SQCOV(I80) = AUXAC*(AUXD-AUXB*GAIN(I))
      GAIN(I) = AUXD*FJ+GAIN(I)
      I80     = I80+1
   80 CONTINUE
  100 CONTINUE
      TRP     = 0.
      DO 120 K= 1,N
      DO 120 J= 1,M
      AB(K,J) = AB(K,J)+GAIN(J)*ERR(K)/SIGM2J
  120 CONTINUE
      DO 130 I= 1,(M*M+M)/2
  130 TRP     = TRP+SQCOV(I)*SQCOV(I)
      RETURN
      END
```

```
C
      SUBROUTINE POLIN(M,N,H,B,Q,NDEG,NR)
      REAL A(10,30),G(10,30,5,5),H(10,5,5),P(10,30,5,5),
     $R(10,30,5,5),Q(15,5,5),B(30)
C     ************************************************
C     *                                              *
C     * N IS THE POLYNOMIAL DEGREE          :INPUT   *
C     * M IS THE MATRIX ORDER (MXM)         :INPUT   *
C     * H IS THE MATRIX TO BE INVERTED (B)  :INPUT   *
C     * B IS THE VECTOR OF DET(B) COEFF     :OUTPUT  *
C     * Q IS THE ADJOINT(B) MATRIX          :OUTPUT  *
C     * NDEG IS THE DEGREE OF DET(B)        :OUTPUT  *
C     * NR IS THE DEGREE OF ADJ(B)          :OUTPUT  *
C     *                                              *
C     ************************************************
      CALL SET1(A,B,G,P,R)
C
C     THIS INSTRUCTION FITS THE INDEX OF THE MATRICES
      DO 60 J=1,N+1
      CALL EQUAL(H,G,M,J)
      CALL TRAZA(G,1,J,M,T)
      A(1,J)    =-T
      DO 60 I1 = 1,M
      DO 60 I2 = 1,M
      ALFA     = 1.
      IF(I2.NE.I1) ALFA = 0.
      R(1,J,I1,I2) = G(1,J,I1,I2)+ALFA*A(1,J)
   60 CONTINUE
      DO 120 I  = 2,M
      DO 70 I1 = 1,N+1
      DO 70 I2 = 1,N*(I-1)+1
   70 CALL PROD(H,R,P,I,M,I1,I2)
C     * THIS CALL PRODUCES P(I2,I2),I1=1,N+1,I2=1,N(I-1)+1*
C     * THE P(I,J) ARE MATRICES IN A PARTITIONED MATRIX   *
      DO 110 L1 = 1,M
      DO 110 L2 = 1,M
      DO 100 J  = 1,N*I+1
      G(I,J,L1,L2) = 0.
      DO 90 K   = 1,N+1
      L         = J-K+1
      IF(L.LT.1) GO TO 90
      IF(L.GT.N*(I-1)+1) GO TO 90
      G(I,J,L1,L2) = G(I,J,L1,L2)+P(K,L,L1,L2)
      CALL TRAZA(G,I,J,M,T)
      A(I,J)    =-T/I
      IF(M.LT.3) GO TO 90
      DO 80 K1= 1,M
      DO 80 K2= 1,M
      ALFA     = 1.
      IF(K2.NE.K1) ALFA = 0.
   80 R(I,J,K1,K2) = G(I,J,K1,K2)+ALFA*A(I,J)
   90 CONTINUE
```

```
 100  CONTINUE
 110  CONTINUE
 120  CONTINUE
C     *                                                          *
C     * THE LAST BLOCK OF INSTRUCTIONS CALCULATE R(I,J).FOR*
C     * I=M,WE GET THE MATRICES OF THE ADJ(H) EXPANSION    *
C     *                                                          *
      F        = (-1.)**(M+1)/M
      DO 130 K=1,N*M+1
      CALL TRAZA(G,M,K,M,T)
 130  B(K)    = F*T
C     * B(I) ARE THE PARAMETERS IN THE  EXPANSION :        *
C     * DET(H) = B(1)+B(2)*S+...+B(NM+1)*S**NM
      ISIGN     = (-1)**(M+1)
      IF(ISIGN.EQ.-1)THEN
      DO 140 I   = 1,N*(M-1)+1
      DO 140 J   = 1,M
      DO 140 K   = 1,M
 140  R(M-1,I,J,K) =-R(M-1,I,J,K)
      ENDIF
      DO 150 I   = 1,N*(M-1)+1
      DO 150 J   = 1,M
      DO 150 K   = 1,M
 150  Q(I,J,K)   = R(M-1,I,J,K)
C     * FILTER FOR NR *
      NR         = N*(M-1)
      N1         = NR
      IF(M.LT.3) GO TO 520
      DO 510 I   = N1,1,-1
      DO 500 J   = 1,M
      DO 500 K   = 1,M
      IF(ABS(Q(I+1,J,K)).GT.1.E-6) GO TO 520
 500  CONTINUE
 510  NR         = NR - 1
 520  CONTINUE
C     * FILTER FOR TAIL ZERO COEFFICIENTS *
      NDEG       = N*M
      CALL XMXIM(NDEG,B,XMX)
      N2         = NDEG
      DO 600 L   = N2,1,-1
      IF(ABS(B(L+1)/XMX).GT.3.E-3) GO TO 610
      NDEG       = NDEG - 1
 600  CONTINUE
 610  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SET1(A,B,G,P,R)
C
      REAL*4 A(10,30),G(10,30,5,5),H(10,5,5),P(10,30,5,5),
     $R(10,30,5,5),B(30)
C
      DO 10 I=1,10
      DO 10 J=1,30
      A(I,J) = 0.
      B(J)   = 0.
      DO 10 K=1,5
      DO 10 L=1,5
      G(I,J,K,L) = 0.
      P(I,J,K,L) = 0.
   10 R(1,J,K,L) = 0.
      RETURN
C
      SUBROUTINE CNTROL(M,N,NAM,A,B,DY,EPS,E,U,ZZ)
C
C     ***************************************************************
C     *                                                             *
C     * PROGRAM FOR CALCULATING THE CONTROL MATRIX GC BY            *
C     * THE INVERSION OF THE MATRIX B.THE MATRICES A AND B          *
C     * ARE PROVIDED BY THE ESTIMATION ALGORITHM.                   *
C     * THE MAXIMUM DIMENSION OF MATRIX B IS M = 5 AND THE          *
C     * MAXIMUM DEGREE OF THE POLYNOMIAL IS N = 10                  *
C     *                                                             *
C     ***************************************************************
C
      COMPLEX XS(20),VS(20),XTEMP
      REAL APOL(21), BPOL(21),CPOL(21),A(10,5,5),B(10,5,5),
     $QN(15,5,5),U(5),XK(5),EPS(5),E(5,20),XX(15,5,5),ZZ(5)
     $,Q(15,5,5),AUX(21),DY(5,20)
      INTEGER D(5)
      COMMON/FIVE/ ERROR
      COMMON/SIX/ P
      COMMON/SEV/ISW
C     * N IS THE POLYNOMIAL DEGREE          *
C     * M IS THE MATRIX ORDER (MXM)         *
C     * B IS THE NUMERATOR MATRIX OF G#     *
C     * A IS THE DENOMINATOR MATRIX OF G#   *
C     * U IS THE CONTROL VECTOR             *
      IF(SQRT(ERROR).LT.0.000500.AND.ISW.EQ.1) GO TO 50
      CALL POLIN(M,N,B,APOL,Q,NDEG,NR)
C
C     * Q IS THE MATRIX ADJOINT(B),APOL IS THE VECTOR OF   *
C     * COEF.FOR THE POLYNOMIAL DET(B),NDEG IS THE DEGREE  *
C     * OF DET(B) AND NR IS THE DEGREE OF ADJ(B)           *
C
      IF(NDEG.EQ.0) THEN
      IF(APOL(1).EQ.0.)THEN
      WRITE(6,130)
  130 FORMAT(12X,'DETERMINANT OF B IS IDENTICALLY NULL ')
```

```
      STOP
      ENDIF
      ENDIF
      IF(N.EQ.1)THEN
      INDEX  = 5
      APOL(1)= APOL(3)
      ANDEG  = 1
      SUM    = 1.
      ND     = 2
      NN     = 0
      DO 124 I  = 1,M
      DO 124 J  = 1,M
 124  QN(1,I,J) = Q(2,I,J)
      GO TO 230
C     * FILTER FOR DETERMINING ND *
      DO 132 I = 1,NDEG+1
  132 AUX(I) = ABS(APOL(I))
      CALL XMX(NDEG+1,APOL,XMAX)
      NV     = NDEG
      L      = 1
  150 CONTINUE
      IF(ABS(APOL(1))/XMAX.GT.1.E-3) GO TO 170
      XS(L)  = (0.,0.)
      NDEG   = NDEG - 1
      DO 160 J = 1,NDEG+1
  160 APOL(J)  = APOL(J+1)
      L      = L + 1
      IF(L.LT.NV+1) GO TO 150
  170 CONTINUE
C
      CALL ROOTS(APOL,NDEG,VS,NPLUS)
C
      DO 180 K = 1,NDEG
  180 XS(NV-NDEG+K) = VS(K)
  190 ND     = NV - NDEG
      NMIN   = NDEG - NPLUS
C
C     * XS ARE THE ROOTS OF DET(B),CPOL IS THE VECTOR OF  *
C     * COEF.FOR DPLUS AND NPLUS IS THE DEGREE OF DPLUS   *
C
   50 CONTINUE
      IF(NPLUS.EQ.0.AND.ND.EQ.0) INDEX = 1
      IF(NPLUS.EQ.0.AND.ND.NE.0) INDEX = 2
      IF(NPLUS.NE.0.AND.ND.EQ.0) INDEX = 3
      IF(NPLUS.NE.0.AND.ND.NE.0) INDEX = 4
  230 CONTINUE
C     SET THE FILTER TUNING PARAMETERS (ALPHA(I))
      XK(1)   = 0.65
      XK(2)   = 0.65
      DO 242 I = 1,M
  242 ZZ(1)    = XK(1)*E(I,1) + (1.-XK(I))*EPS(I)
      GO TO(1,2,3,4,5) INDEX
```

```
    1   NN        = NR
        ANDEG     = 1.
        SUM       = 1.
        CALL SERVO(XX,Q,A,EPS,E,APOL,NDEG,M,NN,NAM,DY,U,ZZ,SUM
       $,ANDEG)
        GO TO 290
    2   CALL DIAG(M,NR,ND,Q,D)
        ANDEG     = 1.
        SUM       = 1.
        CALL BMIN(M,NR,ND,D,Q,QN,NN)
        CALL SERVO(XX,QN, A,EPS,E,APOL,NDEG,M,NN,NAM,DY, U,ZZ,
       $SUM,ANDEG)
        GO TO 290
    3   CALL COEF(VS,BPOL,CPOL,NPLUS,NMIN)
        NN        = NR
        SUM       = 0.
        DO 250 I = 1,NPLUS+1
  250   SUM       = SUM + CPOL(I)
        ANDEG     = APOL(NDEG+1)
        CALL SERVO(XX,Q,A,EPS,E,BPOL,NMIN,M,NN,NAM,DY,U,ZZ,SUM
       $,ANDEG)
        GO TO 290
    4   CALL COEF(VS,BPOL,CPOL,NPLUS,NMIN)
        CALL DIAG(M,NR,ND,Q,D)
        CALL BMIN(M,NR,ND,D,Q,QN,NN)
        SUM       = 0.
        DO 270 I = 1,NPLUS+1
  270   SUM       = SUM + CPOL(I)
        ANDEG     = APOL(NDEG+1)
        CALL SERVO(XX,QN,A,EPS, E,BPOL, NMIN,M,NN,NAM,DY,U,ZZ,
       $SUM,ANDEG)
        GO TO 290
    5   CALL SERVO(XX,QN,A,EPS,E,APOL,NDEG,M, NN,NAM,DY,U, ZZ,
       $SUM,ANDEG)
  330   CONTINUE
        RETURN
        END
C
        SUBROUTINE SET2(Q,QN,VS,XS)
C
        COMPLEX XS(20),VS(20)
        REAL Q(15,5,5),QN(15,5,5)
        DO 30 I=1,15
        DO 30 J=1,5
        DO 30 K=1,5
        Q(I,J,K)  = 0.
   30   QN(I,J,K) = 0.
        DO 50 I=1,20
        VS(I)     = CMPLX(0.,0.)
   50   XS(I)     = CMPLX(0.,0.)
        RETURN
        END
```

```
C
      SUBROUTINE SERVO(XX,G1,A,EPS,E,POL,NDEG,M,NN,NAM,UD,U,
     $ZZ,SUM,ADEG)
C
      REAL G1(15,5,5),A(10,5,5),EPS(5),E(5,20),POL(21),U(5),
     $ZZ(5),XX(15,5,5),UD(5,20),Y(5)
C
C     ************************************************** .
C     * THIS SUBROUTINE CALCULATES THE CONTROL VECTOR U  *
C     * AS      X = EPS+A2*E1+A3*E2+...+ANAM+1*ENAM      *
C     * AND         POL*U    =  X                        *
C     **************************************************
      CALL CLEAR(XX)
      N.sk0;C      = NN + NAM
      CALL IDENT(M,1,XX,G1)
      DO 500 K = 1,M
      DO 500 L = 1,M
      DO 400 I = 2,NC+1
      XX(I,K,L)= G1(I,K,L)
      IF(I.GT.(NN+1)) XX(I,K,L) = 0.
      DO 300 J = 1,I-1
      IF(J.GT.NAM.OR.(I-J).GT.(NN+1))GO TO 300
      DO 300 I1= 1,M
      XX(I,K,L)= XX(I,K,L)+G1(I-J,K,I1)*A(J+1,I1,L)
  300 CONTINUE
  400 CONTINUE
  500 CONTINUE
      ITYPE    = 13
      IF(ITYPE.EQ.1) THEN
      DO 450 L = 1,NC+1
      WRITE(6,420) L
  420 FORMAT(//25X,'GC(',I1,')'//)
      DO 440 J = 1,M
  440 WRITE(6,460)(XX(L,J,K),K=1,M)
  450 CONTINUE
  460 FORMAT(/20X,3(2X,F12.4)/)
      ENDIF
      CALL PROMAT(XX,E,Y,M,NC,ZZ)
      DO 20 I = 1,M
      U(I)    = Y(I)/(ADEG*SUM)
      IF(NDEG.GT.0) THEN
      DO 10 J = 2,NDEG+1
```

```
   10 U(I)     = U(I) - POL(J)*UD(I,J+9)
      ENDIF
      U(I)     = U(I)/POL(1)
   20 CONTINUE
      RETURN
      END
C
      SUBROUTINE BMIN(M,NR,ND,D,Q,QN,N)
C
C     ***************************************************************
C     *                                                             *
C     * BMIN AND DIAG  CALCULATE  THE  PRODUCT  OF  MATRIX          *
C     * ADJ(B) AND DIAGONAL MATRIX D IN ORDER TO COMPENSATE         *
C     * THE MULTIPLE DELAYS                                         *
C     *                                                             *
C     ***************************************************************
      REAL Q(15,5,5),QN(15,5,5),QE(15,5,5,5),ALFA(15)
      INTEGER D(5),DMAX
C
      DO 500 I = 1,15
      DO 500 J = 1,5
      DO 500 K = 1,5
      DO 500 L = 1,5
  500 QE(I,J,K,L) = 0.
      CALL XMAX(M,D,DMAX)
      DO 20 K = 1,NR+1
      DO 20 L = 1,M
      DO 10 I = 1,M
      DO 10 J = 1,M
      ALFA(L) = 0.
      IF(J.EQ.L) ALFA(L) = 1.
   10 QE(K,L,I,J) =Q(K,I,J)*ALFA(L)
   20 CONTINUE
C
C     CALCULATION OF QN
C
      N          = NR+DMAX-ND
      DO 140 I = 1,M
      DO 140 J = 1,M
      DO 130 I1= 1,N+1
      QN(I1,I,J) = 0.
      DO 120 L = 1,M
      K          = ND - D(L) + I1
      IF(K.GT.NR+1) GO TO 120
      QN(I1,I,J) = QN(I1,I,J) + QE(K,L,I,J)
  120 CONTINUE
  130 CONTINUE
  140 CONTINUE
C
      RETURN
      END
```

AUXILIAR PROGRAMS

```
C
      SUBROUTINE DIAG(M,N,ND,Q,D)
C
      REAL*4 Q(15,5,5),AUX(10)
      INTEGER B(1),D(1),P(5,5)
C     ***************************************************
C     *                                                 *
C     * M IS THE MATRIX ORDER                           *
C     * N IS THE POLYNOMIAL MATRIX DEGREE               *
C     * ND IS THE GLOBAL DEAD TIME ASSOCIATED WITH DET(B) *
C     * Q IS THE MATRIX ADJ(B)                          *
C     * D IS THE MATRICIAL PART OF B+                   *
C     * XMAX CALCULATES THE MAXIMUM COMPONENT OF A VECTOR *
C     * (INTEGER)                                       *
C     * XMX CALCULATES THE MAXIMUM COMPONENT OF A VECTOR  *
C     * (REAL)                                          *
C     *                                                 *
C     ***************************************************
      DO 300 J = 1,M
      DO 200 I = 1,M
      P(I,J)   = 1000
      DO 100 K = 1,N+1
  100 AUX(K)   = ABS(Q(K,I,J)
      CALL XMX(N+1,AUX,QMAX)
      DO 110 L = 1,N+1
      IF(AUX(L)/QMAX.GT.0.01) THEN
      P(I,J)   = L-1
      GO TO 200
      ENDIF
  110 CONTINUE
  200 CONTINUE
      DO 220 I1=1,M
  220 B(I1)    = MAX(0,ND-P(I1,J))
      CALL XMAX(M,B,D(J))
  300 CONTINUE
      RETURN
      END
C
      SUBROUTINE SHIFT(N1,N2,X,J,X1)
C
      REAL*4 X(5,20)
      N3          = N1+1
      DO 10 I     = N2,N3,-1
      X(J,I)      = X(J,I-1)
  10  CONTINUE
      X(J,N1)     = X1
      RETURN
      END
```

```
C
      SUBROUTINE  PRBS(N,X,PS)
C
      REAL*4  X(N)
      PS      = -X(1)*X(N-1)
      DO 10 I = 1,N-1
      X(I)    = X(I+1)
   10 CONTINUE
      X(N)    = PS
      RETURN
      END
C
      SUBROUTINE PROMAT(MATRIX,XIN,XOUT,M,N,EPS)
      REAL MATRIX(15,5,5),XIN(5,20),XOUT(5),EPS(5)
C     ************************************************************
C     *                                                          *
C     * THIS SUBROUTINE CALCULATES THE PRODUCT OF A REAL         *
C     * MATRIX M BY A REAL VECTOR XIN GIVING THE PRODUCT         *
C     * VECTOR XOUT                                              *
C     *                                                          *
C     *                                                          *
C     ************************************************************
      DO 100 J = 1,M
      XOUT(J)  = 0.
      DO 100 L = 1,M
  100 XOUT(J)  = XOUT(J) + MATRIX(1,J,L)*EPS(L)
      IF(N.GT.0) THEN
      DO 200 I = 2,N+1
      DO 200 J = 1,M
      DO 200 K = 1,M
      XOUT(J)  = XOUT(J) + MATRIX(I,J,K)*XIN(K,I-1)
  200 CONTINUE
C     WRITE(6,199)(J,XOUT(J),J=1,M)
  199 FORMAT(1X,'FROM PROMAT XOUT(',I1,') = ',2X,F20.5)
      ENDIF
      RETURN
      END
C
      SUBROUTINE ROOTS(B,NDEG,XS,NPLUS)
C
C     ************************************************************
C     *                                                          *
C     * THIS PROGRAM IS AN IMPROVED VERSION OF POLYROOT          *
C     * THE NUMBER OF ITERATIONS AND TOLERANCE ARE FIXED         *
C     * THE ROOTS ARE FREE OF RESIDUALS AND THEY ARE             *
C     * ORDERED ACCORDING TO INCREASING ABSOLUTE VALUE           *
C     * READY TO BE SEPARATED INTO STABLE AND UNSTABLE           *
C     * SETS. POLYROOT IS DUE TO DR. A. CORRIPIO TO WHOM         *
C     * I AM VERY GRATEFUL.                                      *
C     ************************************************************
```

```
      COMPLEX XS(20),XTEMP
      REAL D(20),TEMPR,TEMP,CPOL(21),B(21)
C
C
      DO 8 I = 1,20
    8 D(I)   = 0.
   10 CONTINUE
      MAXIT  = 65
      RTOL   = 1.E-6
      NC = NDEG + 1
C
      DO 20 I=1,NDEG
   20    XS(I) = (0.,0.)
C
C        EVALUATE ROOTS OF POLYNOMIAL BY MULLER'S METHOD
C
      CALL XMULL(B,NDEG, XS, MAXIT, RTOL)
      DO 152 K  = 1,NDEG
      PR        = REAL(XS(K))
      PI        = AIMAG(XS(K))
      IF(ABS(PI).LT.1.E-5) THEN
C     XTEMP     = PR
C     XS(K)     = XTEMP
      XS(K)     = CMPLX(PR,0.)
      ENDIF
      IF(ABS(PR).LT.1.E-5) THEN
      XS(K)     = CMPLX(0.,PI)
      ENDIF
      IF(ABS(PR).LT.1.E-5.AND.ABS(PI).LT.1.E-5) THEN
      XS(K)     = (0.,0.)
      ENDIF
  152 CONTINUE
      DO 142 J = 1,NDEG
  142 D(J)     = CABS(XS(J))
      CALL ORDER(D,XS,NDEG,NPLUS)
      RETURN
      END
C
      SUBROUTINE XMULL(A,N, XS, MAXIT, RTOL )
C
C  PURPOSE - TO COMPUTE APPROXIMATIONS OF THE ROOTS OF
C               NONLINEAR EQUATIONS
C
C  REF:  CONTE AND DE BOOR, "ELEMENTARY NUMERICAL ANALYSIS",
C         3RD ED., MCGRAW-HILL, NEW YORK, 1980, PP. 120-127.
C
C       METHOD:  MULLER'S METHOD OF QUADRATIC INTERPOLATION
C
C       VARIABLES IN ARGUMENT LIST
C
C     VARIABLE    TYPE  I/O  DIMENSION  DESCRIPTION
```

```
C
C       N              I     I      -       NUMBER OF ROOTS TO BE
C                                            COMPUTED
C       XS             C    I/O     ARRAY CONTG. THE INITL.APPROX.
C                                     AND RETG. THE FINAL VALUES
C       MAXIT          I     I      -MAXIMUM NUMBER OF ITERATIONS
C       RTOL           R     I      -    RELATIVE ERROR TOLERANCE
C                                          ON THE ROOTS
C     FUNCTN           E     I      -  NAME OF SUBROUTINE USED TO
C                                          EVALUATE THE FUNCTION
C
C          SUBPROGRAMS CALLED
C
C             FUNCTN - TO EVALUATE THE NONLINEAR FUNCTION
C             DEFLAT - TO DEFLATE THE FUNCTION
C
      COMPLEX XS(N), H, X, FX,F,DFP,FP, LAMBDA, DF, DFPL,
     &DELTA,TOP,G,RAD,BOT
      REAL A(21)
C
C          GUARD AGAINST ZERO OR NEGATIVE ERROR TOLERANCE
C
      ETOL   = AMAX1( RTOL, 1.E-6 )
C
C          START OF LOOP TO EVALUATE N ROOTS
C
      DO 100 I=1,N
         NEVAL = 0
C
C    EVALUATE AND DEFLATE FUNCTION AT FIRST APPROXIMATION
C
 10       H       = 0.5
          X       = XS(I) + H
          NEVAL   = NEVAL + 1
          CALL POLY(A,N,X, FX )
          CALL DEFLAT( I, X, FX, F, XS, IFLG )
          IF( IFLG .NE. 0 ) GOTO 10
C
C    EVALUATE AND DEFLATE FUNCTION AT SECOND APPROXIMATION
C
          DFP    = F
          X      = XS(I) - H
          NEVAL = NEVAL + 1
          CALL POLY(A,N, X, FX )
          CALL DEFLAT( I, X, FX, F, XS, IFLG )
          IF( IFLG .NE. 0 ) GOTO 10
C
C             INITIALIZATION OF ITERATIVE CALCULATION
C
          FP      = F
          DFP     = FP - DFP
          X       = XS(I)
```

```
          LAMBDA = - 0.5
          NEV    = NEVAL + 1
C
C            ITERATIVE CALCULATION OF THE ROOT
C
          DO 30 NEVAL=NEV,MAXIT
C
             CALL POLY(A,N,X, FX )
             CALL DEFLAT( I, X, FX, F, XS, IFLG )
             IF( IFLG .NE. 0 ) GOTO 10
C
C               COMPUTE NEXT ESTIMATE OF ROOT
C
             DF     = F - FP
             DFPL   = DFP * LAMBDA
             DELTA  = 1. + LAMBDA
             TOP    = - 2. * F * DELTA
          G       = ( DELTA + LAMBDA ) * DF - LAMBDA * DFPL
          RAD     = CSQRT( G**2+2.*TOP*LAMBDA * ( DF - DFPL) )
          BOT     = G + RAD
          IF( REAL(G)*REAL(RAD)+AIMAG(G)*AIMAG(RAD) .LT. 0.)
C            BOT = G - RAD
             LAMBDA = TOP
             IF( CABS(BOT) .NE. 0. ) LAMBDA = TOP / BOT
C
             FP  = F
             DFP = DF
             H   = H * LAMBDA
             X   = X + H
C
C               CHECK FOR CONVERGENCE
C
             IF( CABS(H)  .LT. ( ETOL * CABS(X) ) ) GOTO 100
C
 30          CONTINUE
C
C               REACHED MAXIMUM NUMBER OF ITERATIONS
C
          WRITE( 6, 110 ) NEVAL, I, X, F
          WRITE( 4, 110 ) NEVAL, I, X,F
  110 FORMAT(//11X,'FAILED TO CONVERGE AFTER',I3,1X,
     &EVALUATIONS      *,'FOR ROOT',I3//11X,'ROOT',1P,2G15.6
     &,5X,'F(ROOT)',2G15.6)
C
 100      XS(I) = X
C
       RETURN
C
       END
```

```
C*
      SUBROUTINE DEFLAT( I, X, FX, FDEF, XS, IFLG )
C
C       PURPOSE - TO DEFLATE THE FUNCTION F SO THAT THE SAME
C                 ROOT IS NOT ENCOUNTERED MORE THAN ONCE
C
C        VARIABLES IN ARGUMENT LIST
C
C        VARIABLE  TYPE  I/O  DIMENSION  DESCRIPTION
C
C          I         I     I       -       NUMBER OF ROOT SOUGHT
C          X         C     I       -       CURRENT APPROXIMATION
C          FX        C     I       -         FUNCTION VALUE AT X
C          FDEF      C     O       -         DEFLATED VALUE AT X
C          XS        C     I       -       PREVIOUS ROOTS FOUND
C          IFLG      I     O       - FLAG SET TO ONE IF CURRENT
C                                     ROOT MATCHES PREVIOUS ROOT
C
      COMPLEX X, FX, FDEF, XS(1), BOT
C
      IFLG = 0
      FDEF = FX
      IF( I .LT. 2 ) RETURN
C
C         DEFLATION OF F(X)
C
      DO 10 K=2,I
         BOT    = X - XS(K-1)
         IF( CABS(BOT) .LT. 1E-20 ) GOTO 20
 10      FDEF = FDEF / BOT
      RETURN
C
C         CURRENT APPROXIMATION MATCHES A ROOT.MODIFY AND SET
C         IFLAG
 20   XS(I) = X + 0.001
      IFLG  = 1
      RETURN
C
      END
C*
C*
C*
      SUBROUTINE POLY(A,NDEG,X, P )
C
C         PURPOSE - TO EVALUATE A POLYNOMIAL OF NTH DEGREE
C                   FOR SUBROUTINE MULLER
C
C        METHOD  - NESTED MULTIPLICATION (SYNTHETIC DIVISION)
C
C         NDEG          DEGREE OF THE POLYNOMIAL
C         A(NDEG+1)     COEFFICIENTS OF THE POLYNOMIAL ORDERED
C                       IN ASCENDING POWERS OF X
```

```
C
C         VARIABLES IN ARGUMENT LIST
C
C         VARIABLE  TYPE  I/O  DESCRIPTION
C
C           X          C    I        COMPLEX VALUE AT WHICH THE
C                                  POLYNOMIAL IS TO BE EVALUATED
C           P          C    O   COMPLEX VALUE OF THE POLYNOMIAL
C
      COMPLEX X, P, B
      REAL   A(21)
C
      NC = NDEG + 1
      B  = A(NC)
      DO 10 I=1,NDEG
         K = NC - I
 10      B = B * X + A(K)
      P = B
C
      RETURN
      END
C
C
      SUBROUTINE IDENT(M,N,X,Y)
C
      REAL*4 X(15,5,5),Y(15,5,5)
      DO 10 I  = 1,N
      DO 10 J  = 1,M
      DO 10 K  = 1,M
      X(I,J,K) = Y(I,J,K)
  10  CONTINUE
      RETURN
      END
C
      SUBROUTINE COEF(XS,C,B,NP,NM)
C
C     ************************************************
C     *                                              *
C     * THIS SUBROUTINE CALCULATES THE COEFFICIENTS *
C     * FOR A POLYNOMIAL OF DEGREE N WHEN THE ROOTS *
C     * ARE KNOWN (N IS AN INTERNAL PARAMETER>)      *
C     *                                              *
C     ************************************************
      COMPLEX  XS(20),A(20)
      REAL B(21),C(21)
      B(NP+1)  = 1.
      C(NM+1)  = 1.
      DO 10 I = 1,20
  10  A(I)     = (0.,0.)
      A(1)     = XS(1)
      IF(NP.EQ.1) GO TO 100
```

```
      DO 30 I = 2,NP
      DO 20 J = I,2,-1
   20 A(J)    = A(J) + A(J-1)*XS(I)
   30 A(1)    = A(1) + XS(I)
  100 CONTINUE
      DO 40 I = 1,NP
      B(NP+1-I)= A(I)*(-1)**I
   40 CONTINUE
      IF(NM.EQ.C) RETURN
      DO 110 I = 1,20
  110 A(I)    = (0.,0.)
      A(1)    = XS(1+NP)
      IF(NM.EQ.1) GO TO 200
      DO 130 I = 2,NM
      DO 120 J = I,2,-1
  120 A(J)    = A(J) + A(J-1)*XS(I+NP)
  130 A(1)    = A(1) + XS(I+NP)
  200 CONTINUE
      DO 140 I = 1,NM
      C(NM+1-I)= A(I)*(-1)**I
  140 CONTINUE
C     C(NM+1)  = 1.
      RETURN
      END
C
      SUBROUTINE EQUAL(Y,X,M,L)
C
      REAL*4 X(10,30,5,5),Y(10,5,5)
      DO 10 I1 = 1,M
      DO 10 J1 = 1,M
      X(1,L,I1,J1) = Y(L,I1,J1)
   10 CONTINUE
      RETURN
      END
C
      SUBROUTINE TRAZA(X,I,J,M,T)
C
      REAL*4 X(10,30,5,5)
      T        = 0.
      DO 10 L  = 1,M
   10 T        = T + X(I,J,L,L)
      RETURN
      END
C
      SUBROUTINE PROD(X,Y,P,I,M,I1,I2)
C
      REAL  X(10,5,5),Y(10,30,5,5),P(10,30,5,5)
      DO 10 L = 1,M
      DO 10 J = 1,M
      P(I1,I2,L,J) = 0.
      DO 10 K  = 1,M
   10 P(I1,I2,L,J) = P(I1,I2,L,J)+X(I1,L,K)*Y(I-1,I2,K,J)
```

```
      RETURN
      END
C
      SUBROUTINE ORDER(A,XS,N,NPLUS)
      COMPLEX XS(20),CTEMP
      DIMENSION A(1)
      DO 20 J    = 1,N-1
      ITEMP      = J
      RMIN       = A(J)
      DO 10 I    = J+1,N
      IF(A(I).LT.RMIN) THEN
      RMIN       = A(I)
      ITEMP      = I
      ENDIF
   10 CONTINUE
      TEMP       = A(J)
      A(J)       = RMIN
      CTEMP      = XS(J)
      XS(J)      = XS(ITEMP)
      XS(ITEMP)  = CTEMP
   20 A(ITEMP)   = TEMP
      NPLUS      = 0
      DO 30 I    = 1,N
      IF(A(I).GT.1.00) GO TO 50
      NPLUS      = I
   30 CONTINUE
   50 CONTINUE
      RETURN
      END
C
      FUNCTION WNOIS(ISEED,XMEAN,SD)
C
      CALL RANDU(ISEED,ISEED,P)
      WNOIS    = (P-0.5)*2.0*SD+XMEAN
      RETURN
      END
C
      SUBROUTINE XMAX(M,D,IMAX)
C
      INTEGER D(1)
C,    THIS SUBROUTINE CALCULATES THE MAXIMUM COMPONENT OF A
C     VECTOR D (INTEGER VALUES)
      IMAX  = ABS(B(1))
      ILAST = 1
      DO 100 I = 2, M
      IF(ABS(B(I)).GT.IMAX) THEN
      IMAX  = ABS(B(I))
      ILAST = I
      ENDIF
  100 CONTINUE
      RETURN
      END
```

```
C
      SUBROUTINE XMX(M,B,XMAX)
C
      REAL B(1)
C     THIS SUBROUTINE CALCULATES THE MAXIMUM COMPONENT OF A
C     VECTOR B
      XMAX  = ABS(B(1))
      DO 100 I = 2, M
      IF(ABS(B(I)).GT.XMAX) THEN
      XMAX  = ABS(B(I))
      ENDIF
  100 CONTINUE
      RETURN
      END
```

V I T A

Ivan P.Solar was born in Santiago,chile on April 30 1943. He completd his bachelor in Chemical Engineering from Universidad de Chile,Santiago in 1969 and obtained his master of science degree from Imperial College,University of London in 1975. he joined Universidad Catolica de Chile the year 1976. He is now a candidate for the degree of doctor in Philosophy in the department of Chemical Engineering at Louisiana State University.

He is happily married to Isabel and they have a lovely daughter, Katia.

He enjoys classic music,chess and astronomy.

DOCTORAL EXAMINATION AND DISSERTATION REPORT

Candidate: Ivan Patricio Solar

Major Field: Chemical Engineering

Title of Dissertation: Multivariable Adaptive Control Design Under Internal Model Control Structure

Approved:

Frank R. Groves Jr.
Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Roger W. Gibbs

Danny D. Reible

Armando B. Corripio

R. D. Hussey

Date of Examination:

February 17, 1988